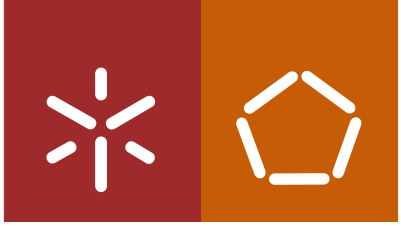
Universidade do Minho
Escola de Engenharia

Bruno Jorge de Sales Gomes Rodrigues

Segurança no Acesso ao Registo Clínico Eletrónico

outubro de 2015

Segurança no Acesso ao Registo Clínico Eletrónico
Bruno Jorge de Sales Gomes Rodrigues
UMinho | 2015



Universidade do Minho
Escola de Engenharia

Bruno Jorge de Sales Gomes Rodrigues

Segurança no Acesso ao Registo Clínico Eletrónico

Dissertação de Mestrado
Mestrado Integrado em Engenharia Biomédica
Ramo de Informática Médica

Trabalho efetuado sob orientação do
Professor Doutor José Manuel Ferreira Machado
e do
Professor Doutor Manuel Bernardo Martins Barbosa

outubro de 2015

Declaração

Nome: Bruno Jorge de Sales Gomes Rodrigues

Endereço eletrónico: bruno.jrodrigues92@gmail.com

Cartão de Cidadão: 14158417

Título da Dissertação: Segurança no Acesso ao Registo Clínico Eletrónico

Orientador: Professor Doutor José Manuel Ferreira Machado
Orientador: Professor Doutor Manuel Bernardo Martins Barbosa

Ano de conclusão: 2015

Designação do Mestrado: Mestrado Integrado em Engenharia Biomédica
Ramo: Informática Médica

Universidade do Minho, __ / __ / __

Assinatura: _____

Agradecimentos

Em primeiro lugar, gostaria de demonstrar, com a máxima sinceridade, a minha gratidão para com os meus orientadores Professor Doutor José Machado e Professor Doutor Manuel Barbosa. Não só por todo o apoio prestado durante este ano, mas também por expressarem o seu interesse no sucesso deste trabalho.

Agradeço aos meus pais todas as oportunidades que me proporcionaram e todo o empenho para que o meu sucesso académico fosse de encontro às minhas expetativas. Não posso deixar de referir a minha irmã, Sofia Rodrigues, que sempre me apoiou e motivou nos momentos mais difíceis.

À minha restante família e aos meus avós em especial, deixo, também, um grande obrigado por demonstrarem sempre o orgulho que têm em mim.

Quanto aos meus amigos, a lista seria extensa para enumerar um a um todos aqueles que foram/são importantes e seria uma injustiça porque de certeza que alguém importante ficaria esquecido. Assim, opto por referir apenas aqueles que tiveram uma influência mais direta no meu percurso académico e fica o sentimento de apreço por todos os outros, sem exceção, aqui não contemplados.

Agradeço ao Gabriel Pinto, Manuel Zamith e Sónia Pereira, companheiros de casa durante o período que passei em Braga. O bom ambiente e a animação reinavam no 1ºC e são momentos que vão perdurar sempre na minha memória.

Ao Luís Torres, colega e amigo, endereço palavras de agradecimento assim como de admiração. Ao desenvolver todos os trabalhos de grupo comigo durante o 4º ano do curso, motivou-me a querer fazer sempre mais e melhor, fazendo-me aprender com os conhecimentos que partilhamos.

A todos os membros do "Gang do Bentinho"agradeço todos os encontros e ocasiões recheados de boa disposição em que estivemos juntos que fizeram o meu percurso nesta Academia mais fácil e, sem dúvida, mais animado.

Ao grupo de amigos "A Seita", à Inês Maia, à Rafaela Oliveira, à Helena Sousa e à Melanie Azeredo, amizades que mantenho dos anos em que estudava no Porto, agradeço a presença constante na minha vida e todo o apoio demonstrado.

Resumo

A visualização de relatórios de Meios Complementares de Diagnóstico e Terapêutica (MCDTs) é importante para uma boa prática médica, no sentido em que estes fornecem indicações importantes sobre o estado de saúde do paciente.

Torna-se, então, crucial o desenvolvimento de uma aplicação que permita a médicos em Centros de Saúde aceder de uma forma segura à informação guardada numa base de dados Hospitalar. Essa necessidade foi encontrada no Centro Hospitalar do Porto, organização para a qual foi implementado este projeto.

Com a presente dissertação, procurou-se, ainda, desenvolver uma análise de segurança ao sistema para que o trabalho final fosse projetado e elaborado de acordo com as reais necessidades do Hospital.

O sistema é apresentado e analisado neste documento. Foram, também, feitos testes à segurança do mesmo, sendo que não foi possível averiguar quanto à sua *performance*.

Abstract

Complementary diagnostic and therapeutic methods (CDTNM) reports are important to a good medical practice as they provide crucial directions about the health's state of a patient.

It is, therefore, critical to develop a software application that allows physicians in local health facilities to access information stored in a Hospital's database in a secure way. This need was spotted in Cento Hospitalar do Porto for which this project was developed.

In this dissertation was implemented a Security Analysis so that the final work was designed and developed according to the real needs of the Hospital.

The system is presented and analyzed in this document. There were also made safety tests to prove its strength. It was not possible to evaluate the system's performance.

Conteúdo

Acrónimos **xx**

1 Introdução **1**

1.1 Enquadramento 1

1.1.1 Conceitos de Segurança de Computadores 2

1.1.2 Modelos de Ataque 3

1.1.3 Criptografia Convencional 4

1.1.4 Criptografia de Chave Pública 6

1.1.5 Acordo de Chaves *Diffie-Hellman* 8

1.1.6 Funções de *Hash* 8

1.1.7 Certificados X.509 10

1.1.8 Secure Sockets Layer (SSL) e Transport Layer Security (TLS) 12

1.1.9 Online Certificate Status Protocol (OCSP) 14

1.1.10 Práticas de Certificação da Entidade Certificadora do Cartão de Cidadão 14

1.2 Motivação 17

1.3 Objetivos 18

1.4 Estrutura do Documento 19

2 Estado da Arte **21**

2.1 Aplicações com Recurso a Métodos Criptográficos 21

2.2 Tecnologia na Saúde 22

2.3 Segurança em Tecnologias na Saúde 23

2.4 Algoritmos Criptográficos 24

3 Metodologia de Investigação e Ferramentas **27**
3.1 Metodologia de Investigação 27
3.2 Ferramentas 29
3.2.1 Programação em Java e *NetBeans* 29
3.2.2 Componente *poreid* 30
3.2.3 *Framework .NET* 30
3.2.4 Base de Dados *Oracle* 31
3.2.5 *Framework Django* 31

4 Análise de Requisitos de Segurança **33**
4.1 Descrição Geral 33
4.1.1 Perspetiva de Produto 34
4.1.2 Restrições Gerais 35
4.1.3 Características de Utilizadores 36
4.2 Análise de Requisitos 36
4.2.1 Visualização de um Relatório por um Utilizador Médico 36
4.2.2 Visualização de Registos por parte de um Utilizador Administrador 37
4.2.3 Registo de um Utilizador Médico por um Administrador 38
4.2.4 Alteração de um Utilizador Médico por um Administrador 39
4.2.5 Eliminação de um Utilizador Médico por um Administrador 40
4.3 Requisitos de Interfaces Externas 41
4.3.1 Interface da Aplicação Visualizador de Relatórios de Métodos Complementares de Diagnóstico e Terapêutica (vRMCDT) 42
4.3.2 Interface da Aplicação Visualizador de Registos (vREG) 44
4.4 Análise de Segurança 47
4.4.1 Definição do Problema de Segurança 47
4.4.2 Objetivos de Segurança 52
4.4.3 Requisitos Funcionais de Segurança 58
4.4.4 Requisitos de Garantia de Segurança 62

4.5 Mecanismos de Segurança 64
4.6 Arquitetura do SSARCE 65
4.6.1 Arquitetura do vRMCDT 65
4.6.2 Arquitetura do Servidor de Relatórios de Métodos Complementares de Diagnóstico e Terapêutica (sRMCDT) . 67
4.6.3 Arquitetura do vREG 69
4.6.4 Comunicação entre Terminais do SSARCE 69
4.7 Política de Certificados do SSARCE 70

5 Resultados **73**
5.1 Aplicação vRMCDT 73
5.2 Comunicação entre o vRMCDT e sRMCDT 80
5.3 Aplicação vREG 82
5.4 Análise SWOT 85
5.4.1 Enquadramento Teórico 85
5.4.2 Análise SWOT do Sistema 86

6 Conclusão **89**
6.1 Contributos 89
6.2 Trabalho Futuro 90

Bibliografia **96**

Apêndices **96**

A Publicações **97**
A.1 Systematic Coronary Risk Evaluation through Artificial Neural Networks based Systems 97
A.2 Improving Quality of MS with M-Health Software 98

B Glossário **101**

Lista de Figuras

1.1 Esquema do Funcionamento de uma Cifra Simétrica (adaptado de [1]). 5
1.2 Esquema de Funcionamento de Criptografia de Chave Pública (adaptado de [1]). 7
1.3 Esquema do Funcionamento de uma Função de *Hash* (adaptado de [1]). 9
1.4 Esquema da Fase de *Handshake* (adaptado de [1]). 13
1.5 Cadeia de Certificação de um Certificado de Autenticação do Cartão de Cidadão 17

2.1 Aplicação do Cartão de Cidadão. 22

3.1 Esquema do Modelo em Cascata (adaptado de [2]). 28

4.1 Esquema do Produto a Desenvolver Inserido no Ambiente em que Funcionará. 34
4.2 Símbolos que Representam os Possíveis Utilizadores do Sistema 35
4.3 Diagrama Representativo da Visualização de um Relatório por um Médico. 36
4.4 Diagrama Representativo da Visualização de Registos de Consultas de Relatórios por parte de um Administrador 38
4.5 Diagrama Representativo do Registo de um Médico por parte de um Administrador 39
4.6 Diagrama Representativo da Alteração dos Dados de um Médico por parte de um Administrador 40

4.7 Diagrama Representativo da Eliminação dos Dados de um Médico por parte de um Administrador 41
4.8 Ecrã Inicial da Aplicação vRMCDT 42
4.9 Ecrã de Pesquisa e Visualização de Relatórios vRMCDT . . . 43
4.10 Ecrã Inicial da Aplicação vREG 44
4.11 Ecrã para Pesquisa de Médicos da Aplicação vREG 45
4.12 Ecrã para Alteração de Dados de Médicos da Aplicação vREG 46
4.13 Ecrã para Inserir um Novo Médico na Aplicação vREG 46
4.14 Símbolo que Representa um Indivíduo Considerado uma Fonte de Risco para o Funcionamento da Aplicação 48
4.15 Esquema Representativo do Acesso de um Intruso a um Terminal com o vRMCDT 49
4.16 Esquema Representativo do Acesso de um Intruso ao Canal de Comunicação entre o vRMCDT e o sRMCDT 49
4.17 Componentes do vRMCDT 66
4.18 Leitor de Cartões Compatível com o Cartão de Cidadão 67
4.19 Componentes do sRMCDT 68
4.20 Componentes do vREG 69
4.21 Protocolos de Comunicação no Sistema Seguro de Acesso ao Registo Clínico Eletrónico (SSARCE) 70

5.1 Janela de Boas-vindas da Aplicação vRMCDT 74
5.2 Janela de Apresentação de Dados do Médico 75
5.3 Janela de Erro, caso não seja possível ler o Cartão de Cidadão 75
5.4 Janela de Introdução de PIN de Autenticação do Cartão de Cidadão 76
5.5 Janela de Pesquisa de Pacientes do vRMCDT 77
5.6 Janela de Apresentação de Exames de um Utente 77
5.7 Lista para Filtragem da Pesquisa dos Exames de um Utente . 78
5.8 Janela de Apresentação dos Detalhes de um Exame 79
5.9 Janela de Visualização de um Relatório de um Exame Complementar de Diagnóstico e Terapêutica 80
5.10 Interface de Autenticação da Aplicação vREG 82

5.11 Interface Inicial da Aplicação vREG. 83
5.12 Interface da Página dos Médicos Registados no Sistema. 83
5.13 Interface dos Registos Referentes à Utilização do vRMCDT. . 84
5.14 Interface de Edição do Registo de um Médico. 84
5.15 Matriz SWOT (adaptado de [3]) 85

Lista de Tabelas

4.1 Métodos de Ataque e as suas Consequências 51
4.2 Relação dos Riscos com os Objetivos de Segurança 55
4.3 Relação das Políticas da Organização com os Objetivos de Segurança 56
4.4 Relação das Suposições com os Objetivos de Segurança 57
4.5 Relação dos Objetivos de Segurança do Sistema com os Requisitos Funcionais de Segurança 61
4.6 Requisitos de Garantia de Segurança 64

Acrónimos

CHP Centro Hospitalar do Porto. 17, 23, 31, 34, 35, 48, 54, 57, 64, 70, 71, 73, 90

DICOM Digital Imaging and Communications in Medicine. 23
DRY Don't Repeat Yourself. 32

EC Entidade Certificadora. 10, 11, 14, 16
EC AuC Entidade de Certificação de Autenticação do Cartão de Cidadão. 14
ENISA European Union Agency for Network and Information Security. 24

GUI Graphical User Interface. 29

HTTP Hypertext Transfer Protocol. 24
HTTPS Hypertext Transfer Protocol Secure. 24

IDE Integrated Development Environment. 29
IETF Internet Engineering Task Force. 13
ITU-T International Telecommunication Union - Telecommunication. 10

LINFO Linux Information Project. 31

MAC Message Authentication Code. 10
MCDT Método Complementar de Diagnóstico e Terapêutica. 2, 17, 23, 91

NIST National Institute of Standards and Techonology. 3

OCSP Online Certificate Status Protocol. ix, 14, 86

SCEE Sistema de Certificação Eletrónica do Estado. 15

SOA Service-Oriented Architecture. 23
SOAP Simple Object Access Protocol. 36
sRMCDT Servidor de Relatórios de Métodos Complementares de Diagnóstico e Terapêutica. xi, xiv, 34–36, 42, 47, 49, 53, 57, 59, 61, 64–67, 69–71, 75, 80, 86
SSARCE Sistema Seguro de Acesso ao Registo Clínico Eletrónico. xiv, 34, 47, 52–54, 59, 64, 69–71, 73, 81, 83, 87
SSL Secure Sockets Layer. ix, 12, 13
SWOT Strenghts, Weaknesses, Opportunities and Threats. 20, 73, 85

TCP Transmission Control Protol. 69
TLS Transport Layer Security. ix, 10, 12, 13, 24, 80

VPN Virtual Private Network. 23
vREG Visualizador de Registos. x, xi, xiv, xv, 34, 35, 38–41, 44–46, 53, 61, 64, 68, 69, 73, 82, 83
vRMCDT Visualizador de Relatórios de Métodos Complementares de Diagnóstico e Terapêutica. x, xi, xiv, xv, 34, 35, 37, 39, 41–43, 47–53, 56, 57, 59–61, 64, 65, 67, 69, 71, 73, 78, 80, 83, 84, 86, 91

Capítulo 1

Introdução

Durante este extenso capítulo, serão apresentados e contextualizados todos os aspetos teóricos relacionados com o projeto desenvolvido. Será ainda explicitada a motivação para a realização deste trabalho e quais os objetivos que se pretendem atingir findo o mesmo. Por fim, relata-se a estrutura deste documento.

1.1 Enquadramento

A recente evolução das Tecnologias de Informação associada à constante mudança dos Sistemas de Saúde faz com que se opte, maioritariamente, por Registos Eletrónicos em Unidades de Saúde. Desta forma, a informação é disponibilizada constantemente aos Profissionais que dela necessitem. [4]

Com estas mudanças, surgem novas preocupações, relativamente à privacidade dos pacientes, já que a informação que se refere é altamente sensível. Tem de haver, então, um equilíbrio entre facilidade de acesso e o controlo no acesso à informação. Deste modo, surgem quatro conceitos envolvidos na proteção de dados relativos à saúde: consentimento, privacidade, integridade e disponibilidade. O primeiro refere-se ao direito e vontade de cada paciente em disponibilizar as suas informações clínicas. A privacidade diz respeito a que informação é disponibilizada para cada paciente e quem a pode visualizar. A integridade é bastante importante no sentido em que atesta se a

informação recebida não foi alterada nem destruída, ou seja, a informação que se visualiza é aquela que foi enviada desde a fonte sem qualquer alteração. Por fim, a disponibilidade diz respeito à capacidade de aceder à informação em qualquer altura de uma forma fiável. [1, 4]

Segundo o Ministério da Saúde, um Método Complementar de Diagnóstico e Terapêutica (MCDT) é uma "designação genérica que engloba exames laboratoriais, imagiológicos, colheita de amostras por meios mais ou menos invasivos, e ainda atos de tratamento variados, realizados em regime ambulatório ou em internamento hospitalar, que têm sido objeto de estatísticas e de comparações para medir a produção dos serviços, e de faturação a terceiros." [5] Estes documentos possuem, então, informação relevante para uma boa prática médica.

Uma aplicação distribuída que permita ter acesso a informação sensível como a presente em relatórios de MCDTs tem, então, de ter em conta os quatro conceitos básicos de segurança referidos anteriormente.

1.1.1 Conceitos de Segurança de Computadores

A Segurança de Computadores contém duas áreas fundamentais. A primeira está relacionada com algoritmos criptográficos e protocolos, enquanto a segunda se refere a segurança de redes e Internet. Estas duas não são mutuamente exclusivas, visto que a segurança de redes e Internet apenas se consegue, utilizando os referidos algoritmos e protocolos [1].

Quanto aos algoritmos e protocolos, considera-se que podem ser divididos em quatro áreas mais específicas: as cifras simétricas, usadas para cifrar blocos de informação de qualquer tamanho, como mensagens, ficheiros, chaves criptográficas ou palavras-passe; as cifras assimétricas, utilizadas para cifrar pequenos blocos de dados como chaves criptográficas, por exemplo; os algoritmos de integridade dos dados, usados para proteger os dados contra alterações não desejadas e, por fim, protocolos de autenticação que servem para autenticar a identidade de certas entidades, utilizando esquemas baseados em algoritmos criptográficos [1].

O campo de segurança de redes e Internet está relacionado com medidas

para deter, prevenir, detetar e corrigir violações ao nível da transmissão de informação pela rede [1]. Noutra perspetiva, segundo o National Institute of Standards and Techonology (NIST), a segurança de computadores é toda a proteção utilizada em sistemas de informação de forma a alcançar os objetivos pretendidos, sendo estes, preservar a integridade, disponibilidade e confidencialidade de recursos de sistemas de informações. Nestes incluem-se *software*, *hardware*, *firmware*, informação e dados [6].

1.1.2 Modelos de Ataque

Existem dois tipos de ataques à segurança de aplicação distribuída em computadores, designados por passivos ou ativos conforme o método utilizado [1].

Ataques Passivos

Um ataque passivo tem como principal característica não alterar ou interromper o bom funcionamento da aplicação. [7]. Existem dois tipos de ataques deste género que é necessário considerar. O primeiro, designado comummente por *Snooping*, define-se como aceder a informação de forma não autorizada. Exemplos disso, serão ler um *e-mail* que aparece no monitor de outra pessoa ou até abrir e ler ficheiros diretamente no computador de outrem [6]. De outra forma, o *Eavesdropping* implica ter acesso ao tráfico que circula na rede e observar esses mesmos pacotes para de alguma forma tentar extrair informação relevante [1].

Estes ataques são de deteção extremamente difícil, visto que quem comunica envia e recebe mensagens sem qualquer alteração e nenhum deles recebe mensagens que não tenham sido enviadas por quem era esperado. Não há, portanto, a perceção de que uma terceira entidade está a visualizar os dados em questão. Devido a isto, para lidar com este tipo de intrusos, é necessário ter em conta mais a prevenção do que a deteção [1].

Ataques Ativos

Um ataque ativo tem como principal objetivo alterar ou destruir os dados que estão a ser trocados no sistema alvo do ataque. Devido a isto, o seu normal funcionamento fica comprometido. Os mesmos podem ser levados a cabo por indivíduos fora da rede do sistema ou até por indivíduos com acesso a uma máquina onde a aplicação é válida, personificando-se por um indivíduo legítimo. Existem quatro diferentes tipos de ataques dignos de registo nesta área. O primeiro refere-se à personificação por um indivíduo legítimo (*masquerade*) em que o atacante se faz passar por um indivíduo com autorização para aceder ao sistema e utilizar os serviços por ele fornecidos. Pode ser executado ao repetir os pacotes que levaram à autenticação de um outro utilizador válido ou, então, ao utilizar o par nome de utilizador e palavra-passe de outrem para obter acesso à aplicação. Num ataque por repetição, o intruso captura pacotes que são trocados pelos terminais válidos do sistema e, de seguida, envia-os de novo para o mesmo destinatário, esperando que este produza uma resposta semelhante para poder obter informação de forma inválida. Por outro lado, num ataque por modificação, o atacante remove, insere ou altera a informação que está a ser trocada, impedindo que a informação enviada pelo utilizador legítimo não chegue ao destinatário ou até chegue alterada. Desta forma, o envio de mensagens produz um efeito não autorizado. Neste tipo de ataques, o intruso pode recorrer ao *Spoofing* em que se mascara num indivíduo legítimo, para tentar ganhar vantagem na troca de informação com o destinatário. Por último, o ataque de negação de serviço é caracterizado por impedir a utilização normal do sistema. Pode ocorrer ao eliminar todas as mensagens enviadas para um certo destinatário ou, então, ao inundar um certo terminal com mensagens de forma a diminuir a sua *performance* (ataque por inundação) [1, 7, 8].

1.1.3 Criptografia Convencional

Para ser possível entender o funcionamento de cifras simétricas (criptografia convencional), é necessário compreender cinco conceitos básicos, apresentados de forma esquemática na figura 1.1 e aqui apresentados [1]:

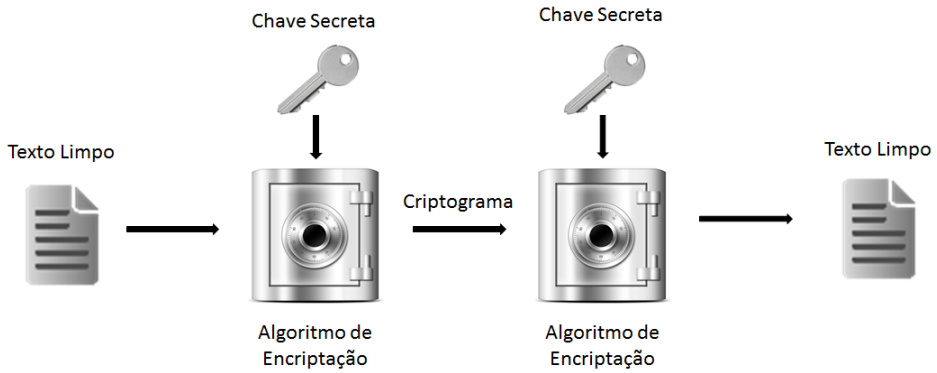


Figura 1.1: Esquema do Funcionamento de uma Cifra Simétrica (adaptado de [1]).

- Texto Limpo: informação ou dados não tratados, que se pretendem esconder e que são o *input* do algoritmo de encriptação.

- Algoritmo de Encriptação: algoritmo que altera o texto limpo de forma a esconder os seus dados originais.

- Chave Secreta: a chave secreta serve também como *input* do algoritmo de encriptação. A saída deste algoritmo depende da chave e é um valor independente do algoritmo e do texto limpo.

- Criptograma: representa a saída do algoritmo de encriptação. Depende da chave secreta e do texto limpo. Aparentemente, ao visualizar um criptograma, este parece completamente aleatório e não revela qualquer informação do texto limpo original.

- Algoritmo de Desencriptação: algoritmo que recebe o criptograma e a chave secreta e, ao fazer o mesmo que o algoritmo de encriptação, mas de forma inversa, produz o texto limpo à saída.

Para uma cifra deste género ser segura, é necessário atingir dois requisitos. O primeiro dita que, se um atacante possuir um certo número de criptogramas e o texto limpo que deu origem a cada um deles, não consegue obter a chave secreta utilizada para cifrar os dados. Por fim, a chave secreta tem de ser

distribuída pelas duas entidades que comunicam de forma segura, pois, se alguém descobrir a chave, o canal de comunicação é considerado inseguro [1].

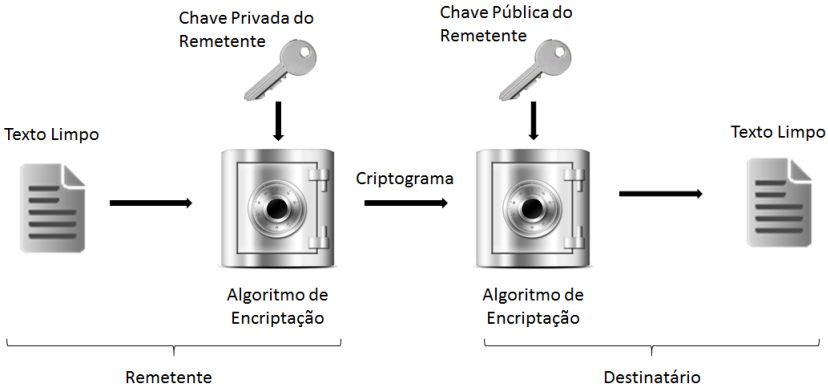
1.1.4 Criptografia de Chave Pública

O conceito de criptografia de chave pública (que utiliza uma cifra assimétrica) [9] apareceu para ultrapassar o problema da distribuição de uma mesma chave secreta entre quem comunica. Ao mesmo tempo, surgiu a necessidade de assinar documentos, tal como uma assinatura convencional em papel, e a criptografia de chave pública veio satisfazer esse requisito [1].

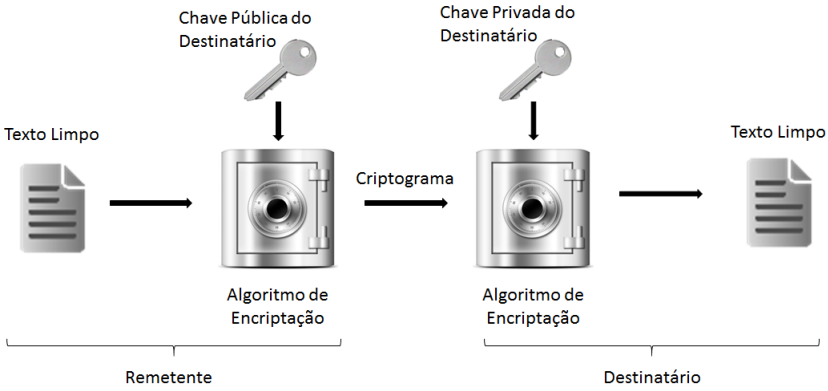
O funcionamento destas cifras baseia-se em cifrar o texto limpo com uma determinada chave e desencriptá-lo com uma chave diferente, mas de certa forma relacionada. Para um bom funcionamento destas cifras, é necessário ser computacionalmente intolerável (em termos de tempo), calcular a chave de desencriptação, tendo como conhecimento base a chave de encriptação e o algoritmo utilizado para gerar as chaves. Alguns algoritmos como o *RSA* permitem que ambas as chaves sejam usadas para encriptação, desde que a outra seja usada no processo inverso (desencriptação) [1].

Como uma cifra simétrica (anteriormente apresentada), uma cifra assimétrica assenta em seis conceitos, que podem ser visualizados nas figuras 1.2a e 1.2b [1]:

- Texto Limpo: informação ou dados não tratados, que se pretendem esconder e que são o *input* do algoritmo de encriptação.
- Algoritmo de Encriptação: algoritmo que altera o texto limpo de forma a esconder os seus dados originais, utilizando a chave pública ou privada.
- Chaves Pública e Privada: par de chaves utilizado para cifrar e desencriptar o texto limpo. É necessário, obrigatoriamente, usar uma delas para cifrar e a outra para obter o texto limpo.
- Criptograma: representa a saída do algoritmo de encriptação. Depende da chave e do texto limpo. Para um mesmo texto limpo, utilizando ou a chave pública ou a privada, obtêm-se criptogramas diferentes.



(a) Encriptação com a Chave Privada.



(b) Encriptação com a Chave Pública.

Figura 1.2: Esquema de Funcionamento de Criptografia de Chave Pública (adaptado de [1]).

- Algoritmo de Desencriptação: algoritmo que recebe o criptograma e, entre a chave pública e a privada, aquela que não foi usada para cifrar os dados de forma a produzir o texto limpo à saída.

Deste modo, para comunicar confidencialmente, utilizando este esquema, é preciso que o destinatário crie um par de chaves e publique a sua chave pública para que todos possam, ter acesso à mesma. Quem quiser comunicar com essa entidade cifra os dados com a chave pública e, desde que a chave privada não tenha sido divulgada, apenas o dono da mesma pode obter o texto limpo a partir do criptograma. Por isso, a comunicação é considerada

segura desde que a chave privada não seja difundida [1].

1.1.5 Acordo de Chaves *Diffie-Hellman*

O propósito do algoritmo de acordo de chaves *Diffie-Hellman* é permitir que dois indivíduos acordem uma chave simétrica pela rede no momento inicial da comunicação. De uma forma simplificada, cada entidade possui uma chave privada e uma chave pública. Ao serem trocadas as chaves públicas, cada sujeito é capaz de computar a mesma chave, utilizando a sua própria chave privada e a chave pública que recebeu [1, 10].

Para o processo ser possível devem, previamente, ser acordados dois valores. Um deles, um número primo (p) e um número inteiro que seja raiz primitiva do primeiro (g). Um dos utilizadores computa um valor (Xa) menor que p e calcula $Ya = gXa \ mod \ q$, que corresponde ao seu valor público. O segundo utilizador faz cálculos semelhantes e ambos trocam as respetivas chaves públicas. Para calcular a chave (K), o primeiro utilizador calcula, $K = Yb \ Xa \ mod \ q$, sendo que o segundo faz o mesmo cálculo, mas utilizando o seu valor privado e o público do outro indivíduo [1].

Um atacante conhece os valores p, g, Ya e Yb. Para calcular a mesma chave, ele necessitaria de calcular Xa ou Xb, utilizando a expressão $Xb = d \ log \ p,g \ (Yb)$. De seguida, obteria a chave da mesma forma que o utilizador que possui o valor Xb. A segurança deste algoritmo recai sobre o facto de ser relativamente fácil calcular módulos de números primos e ser difícil calcular logaritmos discretos. Para números primos grandes, a tarefa é computacionalmente impraticável [1].

1.1.6 Funções de *Hash*

Uma função de *hash* é uma função que recebe um bloco de dados de comprimento variável, como uma mensagem, por exemplo, e retorna um valor de *hash* de comprimento fixo. A principal propriedade destas funções é que, para um grande número de *inputs*, os valores de *hash* são aparentemente aleatórios. Por isso, esta família de funções é usada para verificar a integridade

Figura 1.3: Esquema do Funcionamento de uma Função de *Hash* (adaptado de [1]).

de mensagens, visto que a alteração de apenas um *bit* numa mensagem produz uma mudança significativa no valor de *hash* quando a mesma é aplicada na função. O seu funcionamento pode ser esquematizado pela figura 1.3 [1].

Para o uso em sistema de segurança de computadores, estas funções são, geralmente, designadas funções de *hash* criptográficas. Para estas funções serem consideradas seguras, é necessário que seja computacionalmente inviável encontrar qual a mensagem que gerou um certo valor de *hash* conhecido e descobrir duas mensagens diferentes que gerem o mesmo valor de *hash* [1].

Devido a estas características, estas funções são utilizadas para verificar que, numa mensagem, não houve modificação, inserção, eliminação ou repetição de certos *bits*. Para isso, o emissor envia, para o destinatário, a mensagem propriamente dita e o seu respetivo valor de *hash*, ficando o segundo encarregue de calcular o valor de *hash* para a mensagem que recebeu e comparar se o valor obtido é igual ao valor recebido. Caso isso não se verifique, o destinatário sabe que a mensagem foi alterada ou até mesmo o valor de *hash* [1].

Esta forma de verificar a integridade de mensagens é facilmente contornada por um ataque *man-in-the-middle*, em que o atacante recebe a mensagem enviada pelo remetente, cria uma mensagem totalmente nova e envia para o destinatário original essa mensagem e o respetivo valor de *hash*. Como a verificação da integridade pelo destinatário indicará que não houve alteração da mensagem, ele confia nessa mensagem recebida, apesar de não ter sido a mensagem originalmente enviada. Para resolver este problema, é necessário proteger o valor de *hash* enviado pelo remetente [1].

Comummente, a integridade de mensagens é verificada, utilizando um có-

digo de autenticação de mensagens (Message Authentication Code (MAC)). Uma função MAC recebe uma mensagem e uma chave que as duas entidades partilham e produz um valor de *hash* que é associado a essa mensagem. A verificação é feita de forma convencional, já referida anteriormente, mas utilizando a respetiva chave. Como um qualquer atacante não possui (teoricamente) a chave secreta, não consegue produzir um valor de *hash* válido para cada mensagem [1].

1.1.7 Certificados X.509

Os certificados X.509 são uma recomendação da International Telecommunication Union - Telecommunication (ITU-T) que definem serviços de autenticação para os seus utilizadores através de certificados de chave pública. Cada certificado contém, assim, uma chave pública e é assinado com a chave privada de uma Entidade Certificadora (EC) [1].

Este *standard* é importante, pois é largamente utilizado noutros contextos, nomeadamente no Transport Layer Security (TLS) explicado noutra secção deste documento [1].

A recomendação X.509 é baseada em criptografia de chave pública e recomenda a utilização do algoritmo RSA [1].

O formato geral de um certificado deste género contém informações sobre a entidade a quem pertence o certificado e sobre o certificado em si, a chave pública e a assinatura. Dependendo da versão do certificado (de um a três), existem diferentes atributos, aqui enumerados e explicados [1]:

- Versão: um é a versão padrão, mas poderá ter os valores dois ou três, de acordo com os valores que possa ter a mais.
- Número de Série: um valor inteiro único dentro da Autoridade de Certificação que identifica o certificado.
- Identificação do Algoritmo de Assinatura: algoritmo utilizado para gerar a assinatura juntamente com alguns parâmetros necessários. Esta informação é repetida no campo da assinatura, daí ter quase nenhuma utilidade.

- Nome da Autoridade de Certificação: nome da Autoridade de Certificação que assinou este certificado.
- Data de Validade: possui duas datas, representando, respetivamente, o primeiro e último dias em que o certificado é válido.
- Nome da Entidade: nome da entidade a quem o certificado pertence, ou seja, o certificado em causa certifica a chave pública do utilizador que contém a chave privada correspondente.
- Informação da Chave Pública da Entidade: contém a chave pública, o algoritmo com que a chave deve ser usada juntamente com outros parâmetros necessários.
- Identificação Única da Autoridade de Certificação: campo opcional utilizado para identificar a EC, caso o nome da EC esteja a ser utilizado por outra entidade.
- Identificação Única da Entidade: campo opcional utilizado para identificar a entidade possuidora do certificado, caso o nome da Entidade esteja a ser utilizado por outrem.
- Extensões: as extensões foram acrescentadas na versão três.
- Assinatura: contém o valor de *hash* de todos os outros campos encriptados com a chave da Autoridade de Certificação, para além do identificador do algoritmo utilizado na assinatura.

Na versão dois, foram adicionados os campos de identificação única da EC e da Entidade possuidora do certificado, mas raramente são usados. Na versão três, foram adicionadas extensões ao certificado, para corrigir alguns problemas identificados na versão dois, mas não vão aqui ser explicados visto não terem relevância no trabalho efetuado. Como a assinatura foi gerada, utilizando a chave privada da EC, qualquer utilizador que tenha acesso à chave pública correspondente pode verificar se um certificado assinado pela mesma é válido [1].

1.1.8 Secure Sockets Layer (SSL) e TLS

O SSL é um dos serviços mais utilizados, para garantir segurança a um canal de comunicação na rede. A última versão do SSL é a versão três (SSLv3) e tem pequenas diferenças para o TLS [1].

O SSL é composto por duas camadas de protocolos. Neste documento, serão analisadas as camadas designadas *SSL Record Protocol* e *SSL Handshake Protocol* [1].

A camada *SSL Record Protocol* permite conferir à ligação confidencialidade e integridade de mensagens. A primeira é obtida através de um acordo de uma chave simétrica na fase de *handshake* utilizada para cifrar os dados que vão ser enviados, enquanto a integridade é verificada, utilizando um código de autenticação de mensagem já explicado neste capítulo, usando uma chave também previamente acordada [1].

A camada *SSL Handshake Protocol* é a fase mais complexa deste serviço. É durante este processo que, tanto cliente como servidor se autenticam, negoceiam os algoritmos a utilizar quer para a cifra quer para o código de autenticação de mensagens e acordam as chaves necessárias em tais processos. Antes de qualquer dado ser enviado, utilizando SSL, é efetuado este processo de *handshake* que pode ser visualizado pelo diagrama simplificado da figura 1.4.

Inicialmente, o cliente que deseja comunicar com o servidor envia uma mensagem intitulada *client_hello,* em que remete, entre outros dados, quais os algoritmos criptográficos por ele suportados. Após esta mensagem, o cliente espera por uma mensagem designada *server_hello* em que, para além de outros parâmetros, são especificados os algoritmos que vão ser utilizados e que estavam na lista previamente enviada pelo cliente (fase um da figura 1.4) [1, 11].

Posteriormente, o servidor inicia a fase 2 (figura 1.4), enviando o seu certificado X.509 ao cliente (ou uma cadeia de certificados), os seus valores públicos para o acordo de chaves (se não estiverem definidos no certificado) e uma mensagem, para pedir o certificado correspondente ao cliente com quem estabelece a ligação. Nesta última comunicação, são transmitidos dados sobre

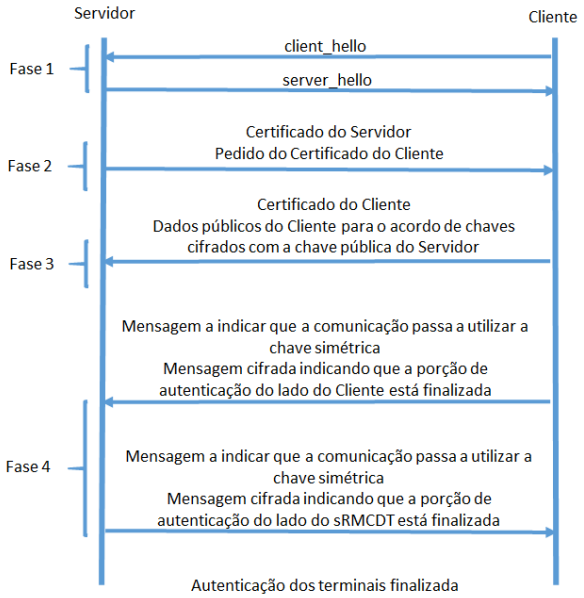


Figura 1.4: Esquema da Fase de *Handshake* (adaptado de [1]).

as autoridades de certificação aceites. É, por fim, endereçada uma mensagem que indica o fim do envio de dados por parte do servidor [1, 11].

No início da fase três (figura 1.4), o cliente verifica o certificado do servidor e envia o seu próprio certificado, caso tenha sido pedida tal informação. De seguida, são remetidos os dados públicos do cliente para o acordo de chaves assinados pela chave pública do servidor e, se o certificado do cliente possuir capacidade de assinatura, é remetida uma mensagem a pedir para o servidor verificar o certificado enviado [1, 11].

Na última etapa (fase quatro da figura 1.4), é onde termina esta camada do protocolo. O cliente envia uma mensagem, mostrando que o mesmo começou a utilizar os algoritmos previamente acordados e uma mensagem de fim do acordo aplicando já esses algoritmos, usada para verificar que as chaves geradas foram as corretas. A resposta a estas duas mensagens são duas mensagens semelhantes. Neste momento, todo o processo fica concluído e pode ser iniciada a comunicação entre cliente e servidor [1, 11].

O TLS é uma norma para uniformizar o SSL por parte da Internet Engineering Task Force (IETF). É muito similar à versão três do SSL previamente descrita. As diferenças consistem em algumas pequenas alterações ao nível

do algoritmo de integridade de mensagens, dos algoritmos disponíveis para acordo de chaves e cifra simétrica, etc [1].

1.1.9 Online Certificate Status Protocol (OCSP)

O OCSP é um protocolo que permite verificar o estado de um certificado (se é válido ou, por algum motivo, foi revogado). A grande vantagem deste método é permitir obter informação mais atualizada, relativamente a listas de revogação de certificados. Para obter a informação, o cliente (entidade que quer validar o certificado) faz um pedido designado *status request* e suspende a aceitação desse certificado até obter uma resposta [12].

Um pedido que utilize este protocolo possui a versão do protocolo que está a ser usada; qual o pedido que está a ser efetuado, qual o certificado de que se quer obter informação e algumas extensões opcionais [12].

A respetiva resposta contém, também, a versão do protocolo; o nome da entidade que respondeu ao pedido; a resposta em si, indicando o estado do certificado em questão; extensões opcionais e a assinatura da resposta. Esta assinatura é gerada, utilizando a chave privada da EC que emitiu o certificado; a chave privada de uma entidade de confiança, sendo que o cliente confia na correspondente chave pública ou pela chave privada de uma entidade designada pela EC do certificado em questão [12].

O estado de um certificado pode ser *good*, indicando que o certificado é válido, atestando que, pelo menos, este não está revogado. Pelo contrário, uma resposta com um estado *revoked* significa que esse certificado foi revogado. Por fim, pode-se obter uma resposta *unknown*, concluindo-se que a entidade a quem se emitiu o pedido não possui informações relativas a esse certificado [12].

1.1.10 Práticas de Certificação da Entidade Certificadora do Cartão de Cidadão

Na infraestrutura de Chave Pública do Cartão de Cidadão estão presentes diversas entidades. A Entidade de Certificação de Autenticação do Cartão

de Cidadão (EC AuC) está presente na hierarquia de confiança do Sistema de Certificação Eletrónica do Estado (SCEE) e é a responsável por emitir, operar, suspender e revogar certificados de autenticação dos cidadãos portugueses [13].

No topo da cadeia de certificados do Estado, encontra-se a EC Raiz do Estado, sendo esta a raiz da infraestrutura de chaves públicas do Estado Português. Este certificado tem as seguintes características [13]:

1. Nome comum: ECRaizEstado
2. Organização: SCEE
3. País: PT
4. Validade: De sexta-feira, 23 de junho de 2006 14:41:27. Até domingo, 23 de junho de 2030 14:41:27

Este certificado encontra-se assinado por um certificado autoassinado denominado *Baltimore CyberTrust Root* [13].

Logo abaixo na cadeia de certificados, encontram-se as ECEstado. A função desta entidade é emitir, suspender e revogar certificados para as suas SubECEstado. Este certificado tem as seguintes características [13]:

1. Nome comum: Cartão de Cidadão 001
2. Unidade Organizacional: ECEstado
3. Organização: SCEE - Sistema de Certificação Eletrónica do Estado
4. País: PT
5. Validade: De sexta-feira, 26 de janeiro de 2007 19:50:41. Até domingo, 27 de maio de 2018 20:00:41

Atualmente, existe um segundo certificado pertencente à ECEstado com os seguintes atributos [13]:

1. Nome comum: Cartão de Cidadão 002

2. Unidade Organizacional: ECEstado
3. Organização: SCEE - Sistema de Certificação Eletrónica do Estado
4. País: PT
5. Validade: De quinta-feira, 30 de maio de 2013 12:15:13. Até sexta-feira, 30 de maio de 2025 12:15:13

Logo abaixo das ECEstado, encontram-se as SubECEstado que prestam o serviço ao utilizador final. Esta EC emite os certificados de autenticação que estão presentes em cada Cartão de Cidadão. Todos os anos e desde 2007, é criado um novo certificado para esta EC, sendo que o último foi o número dez. Os primeiros oito foram assinados pela ECEstado número 001 e os últimos dois foram assinados pela ECEstado 002. Aqui ficam os valores presentes no primeiro certificado da SubECEstado [13]:

1. Nome comum: EC de Autenticação do Cartão de Cidadão 0001
2. Unidade Organizacional: subECEstado
3. Organização: Cartão de Cidadão
4. País: PT
5. Validade: De Segunda-feira, 29 de janeiro de 2007 22:32:58. Até sábado, 30 de março de 2013 22:42:58

O utilizador final possui, então, um certificado de autenticação assinado por uma das SubECEstado. Este certificado possui os seguintes atributos [14]:

- País: PT
- Organização: Cartão de Cidadão
- Unidade Organizacional: Cidadão Português
- Unidade Organizacional: Autenticação do Cidadão

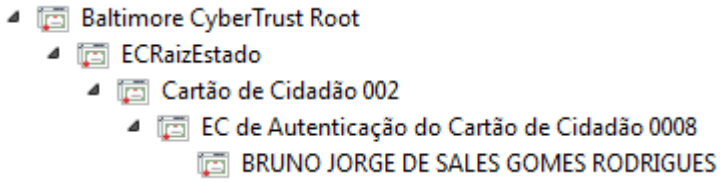


Figura 1.5: Cadeia de Certificação de um Certificado de Autenticação do Cartão de Cidadão

- Nome Comum: <Nome completo do cidadão>
- Apelido: <Nome de família do cidadão>
- Nome: <Nome do cidadão>
- Número: <Identificador único do cidadão>

A figura 1.5 comprova a referida cadeia de certificação desde a sua raiz até ao certificado de autenticação do cidadão.

1.2 Motivação

A visualização de relatórios de MCDT é importante para uma boa prática médica, no sentido em que estes fornecem indicações cruciais sobre o estado de saúde do paciente. É, então, necessário que os mesmo se encontrem disponíveis para quem os requisitou. Infelizmente, médicos que não exerçam na Instituição Hospitalar onde foi realizado o exame/análise não terão acesso ao respetivo relatório. Desta forma, será criada uma ferramenta/aplicação que permita a um Profissional de Saúde aceder a estes dados, até agora restritos. Para isso, terão de ser tidas em conta questões de segurança, utilizando princípios de Criptografia, para assegurar a privacidade do paciente e identificar inequivocamente qual o Profissional que está a aceder a essa informação.

O Centro Hospitalar do Porto (CHP) será a unidade hospitalar onde este projeto será testado.

1.3 Objetivos

Tal como mencionado anteriormente, a presente dissertação tem como objetivo central a criação de uma aplicação distribuída de apoio aos Profissionais de Saúde, para garantir o acesso a relatórios de exames ou análises de uma máquina exterior à Intranet Hospitalar.

Para ser atingido o pleno sucesso neste projeto, optou-se por dividi-lo em pequenos objetivos:

1. Executar uma análise às necessidades do produto:
    - Definir os requisitos funcionais do sistema;
    - Estudar as restrições que têm de ser tidas em conta;
    - Analisar as características de cada tipo de utilizador;
2. Proceder a uma análise de segurança:
    - Identificar o problema de segurança;
    - Aclarar suposições, relativamente aos utilizadores e ambiente do sistema;
    - Especificar quais os riscos e os possíveis ataques;
    - Precisar as políticas da organização a ser tidas em conta;
3. Determinar quais os objetivos de segurança;
4. Enumerar os objetivos de garantia de segurança;
5. Apresentar a arquitetura do sistema:
    - Justificar os componentes do sistema;
    - Elucidar acerca dos protocolos de comunicação entre os mesmos;
6. Desenvolver o sistema proposto;
7. Realizar testes e estudos ao sistema;

1.4 Estrutura do Documento

O presente documento encontra-se estruturado em capítulos, sendo estes: introdução, estado da arte, metodologia de investigação e ferramentas, análise de segurança, resultados e conclusão e trabalho futuro.

Este mesmo capítulo (introdução) enquadra o trabalho que levou ao desenvolvimento do mesmo; apresenta de uma forma teórica todos os conceitos necessários à compreensão do projeto que é apresentado nos capítulos subsequentes; explicita as motivações para a sua realização; enumera os objetivos que se pretendem atingir, findo o projeto e qual a estrutura completa da dissertação. Seguem-se os seguintes capítulos:

Capítulo 2 - Estado da Arte

No capítulo 2, são apresentados alguns dos trabalhos desenvolvidos por outros investigadores, na mesma área deste trabalho ou numa diretamente relacionada, que permitiram enquadrar o presente projeto, para que fosse considerado atual e igualmente inovador. Ao mesmo tempo, são analisados os requisitos dos parâmetros de algoritmos criptográficos, para uma aplicação ser considerada segura.

Capítulo 3 - Metodologia de Investigação e Ferramentas

No capítulo 3, é referida e explicada a metodologia de investigação utilizada no processo de desenvolvimento do sistema proposto. De seguida, apresentam-se com uma breve descrição todas as ferramentas utilizadas. Entre elas, programação em Java e *NetBeans*, componente *poreid*, *Framework .NET*, base de dados *Oracle* e *Framework Django*.

Capítulo 4 - Análise de Requisitos de Segurança

O capítulo 4 apresenta a proposta de sistema a desenvolver no âmbito desta dissertação. São descritas quais as funções que a aplicação deve per-

mitir executar e apresentam-se de forma grosseira as interfaces gráficas que devem ser implementadas no produto final. Mais importante, ainda, é a análise de segurança conduzida e apresentada nesse capítulo, pois permite perceber quais os objetivos de segurança que se pretendem atingir. Dessa forma, é possível projetar o sistema final de modo mais eficiente.

Capítulo 5 - Resultados

No capítulo 5, mostra-se através de imagens o produto final, analisando-se, ao mesmo tempo, como funciona a comunicação entre as partes do sistema. Dessa maneira é possível mostrar que tipo de segurança foi implementada. No final dessa secção, é, ainda, apresentada uma análise Strenghts, Weaknesses, Opportunities and Threats (SWOT) ao sistema.

Capítulo 6 - Conclusão

O capítulo 6 e último procura, de uma forma sucinta, apresentar as conclusões obtidas no final do trabalho e verificar quais dos objetivos inicialmente propostos foram cumpridos. Posteriormente, fazem-se algumas sugestões de trabalho futuro de forma a melhorar a aplicação.

Capítulo 2

Estado da Arte

Ao longo deste capítulo, irão ser feitas referências ao trabalho que tem sido desenvolvido recentemente relativo a algoritmos criptográficos. Da mesma forma, irão ser abordadas e analisadas aplicações de acesso à informação clínica, visto que é o âmbito do presente projeto.

Ao mesmo tempo, serão apresentadas aplicações que fazem uso do Cartão de Cidadão.

Por outro lado, estudam-se também quais os parâmetros mínimos considerados seguros para a utilização dos algoritmos criptográficos apresentados no capítulo 1.

2.1 Aplicações com Recurso a Métodos Criptográficos

O Estado Português disponibiliza a todos os seus cidadãos uma aplicação designada Aplicação Cartão de Cidadão, porque o Cartão de Cidadão é muito mais do que uma identificação física, é também um documento eletrónico que possibilita a realização de várias operações sem necessidade de interação presencial. Neste sentido, com este documento é possível a autenticação eletrónica segura, a geração de assinaturas eletrónicas que têm o mesmo valor legal de uma assinatura manuscrita e a alteração da morada [15]. Desta forma, todos os portugueses detentores do cartão podem executar es-

tas funcionalidades numa aplicação visualizada na figura 2.1. Esta aplicação é utilizada neste projeto de dissertação, já que a biblioteca poreid descrita na secção 3.2.2 faz uso desta aplicação como *middleware*.

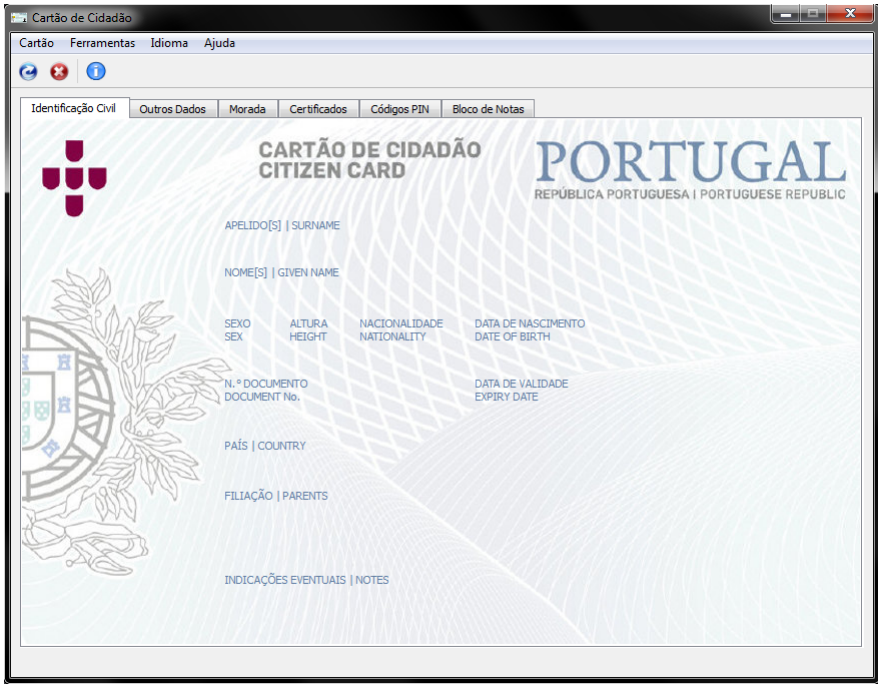


Figura 2.1: Aplicação do Cartão de Cidadão.

2.2 Tecnologia na Saúde

Muitos profissionais de saúde estão a adotar aplicações distribuídas para tornar os seus procedimentos mais eficientes. Estas aplicações são desenvolvidas para diferentes dispositivos e com finalidades diferentes. Por exemplo, existem aplicações desenvolvidas para computadores fixos [16] ou para dispositivos móveis [17] e aplicações baseadas em agentes [18] ou na *Web* [19].

As aplicações utilizadas na área da Saúde, atualmente, podem ter objetivos e interesses altamente variados. Enquanto umas têm como interesse a assistência a doentes com determinadas doenças, como as referidas por Jean, F. *et al.* [17] e Bardram, Jakob *et al.* [20] outras contemplam o acesso a informação médica e suporte à decisão médica, como os apresentados por

Oliveira, R. *et al.* [21] e por Pereira, S. *et al.* [22]. Por outro lado, existem, ainda, aplicações que permitem prever certos acontecimentos, sendo o caso da proposta por Pereira, S. *et al.* [23].

No caso mais particular deste projeto de dissertação, o interesse recai sobre aplicações que possibilitam a consulta de informações clínicas.

A aplicação desenvolvida por Pereira, A. *et al.* [24] faculta a consulta de relatórios de MCDT no CHP através de um qualquer dispositivo *Android*. Esta aplicação é baseada no padrão Service-Oriented Architecture (SOA), ou seja, uma aplicação que utiliza serviços intermediários para o acesso à base de dados. Em termos de segurança, esta aplicação utiliza, apenas, um sistema de autenticação de utilizadores relativos ao seu nome de utilizador e palavra-passe.

Outras aplicações como o *PatientKeeper* permitem a visualização do registo clínico eletrónico do Hospital em que está integrado, tanto através de um Virtual Private Network (VPN) como através da Internet e em diversos dispositivos com características diferentes (fixos ou móveis) [25].

2.3 Segurança em Tecnologias na Saúde

A maioria das aplicações utilizadas na área da Saúde tem em conta requisitos fundamentais de segurança, para assegurar a privacidade dos doentes.

Os autores *Herveg, J. et al.* [26] descrevem os requisitos legais de um serviço para aceder a dados pessoais de doentes. A infraestrutura proposta e analisada segue os requisitos imposto nos *X.509 Internet Drafts and standards* [27], utilizando certificados X.509 para os processos de autenticação, identificação e autorização tanto de máquinas como de pessoas. Para além disso, este serviço é *Web-Service oriented*, ou seja, orientado para Serviços *Web*.

Os investigadores *Chervenak, A et. al* [28] descrevem uma aplicação para acesso a imagens de métodos complementares de diagnóstico e terapêutica no formato Digital Imaging and Communications in Medicine (DICOM). Este sistema utiliza certificados X.509, entre outras tecnologias, para autenticação de utilizadores, contribuindo para a privacidade dos dados sensíveis.

Os investigadores *Gelogo, E.* [29] e *Kim, H.* propuseram um *design* para uma gestão segura de dados hospitalares eletrónicos. Nesta aplicação, existe troca de dados hospitalares para profissionais de saúde ou um outro consumidor através da Internet. Para impor segurança, são utilizados esquemas criptográficos que utilizam assinaturas digitais com recurso a certificados X509 e funções de *hash*. Na mesma proposta, inicialmente é necessário garantir permissões de acesso ao utilizador e depois autenticar os dois terminais que comunicam através dos referidos métodos.

O sistema distribuído para dispositivos médicos apresentado por *Gregorczyk, D. et. al* [30] é um sistema orientado a serviços *Web* para consulta de dados de doentes. Como tal, a segurança do sistema foi um dos pontos tidos em conta aquando da sua construção, utilizando certificados X.509 para autorização de utilizadores. Na comunicação, faz uso de Hypertext Transfer Protocol Secure (HTTPS), ou seja, utiliza TLS juntamente com o Hypertext Transfer Protocol (HTTP).

2.4 Algoritmos, Tamanhos de Chaves Criptográficas e Parâmetros

Segunda a European Union Agency for Network and Information Security (ENISA), a cifra por blocos que deve ser utilizada atualmente, com garantia de segurança é o AES. O tamanho do bloco mínimo produzido pelo algoritmo deverá ser 128 *bits*. Tamanhos de blocos maiores aumentam a segurança, mas também o tempo de computação do algoritmo. Este algoritmo é, ainda, considerado bom para uso de aplicações no futuro (de acordo com o documento, é expectável que se mantenha seguro durante mais dez a cinquenta anos) [31].

Segundo a mesma entidade, para uma função de *hash* ser segura, o tamanho do seu *output* deveria ser, pelo menos, 160 *bits*, enquanto para aplicações novas ou futuras, o seu comprimento já deva ser 256 *bits*. Quanto aos algoritmos, são considerados aceitáveis, para um uso a longo prazo, os seguintes: família de algoritmos SHA-2 (SHA-256, SHA-384 e SHA-512); SHA3 e Whirlpool. Para aplicações atuais, algoritmos como o SHA-1 são

considerados seguros [31].

Quanto a algoritmos de criptografia de chave pública, é considerado seguro, atualmente, o algoritmo RSA com um tamanho de chave de 1024. Para aplicações para uso a maior longo prazo, assume-se que um tamanho de chave superior a 3072 *bits* é aceitável [31].

Relativamente ao acordo de chaves, o algoritmo de *Diffie-Hellman* com parâmetros com comprimento superior a 160 *bits* são seguros, para aplicações a ser utilizadas no momento. Para obter segurança a mais longo prazo, é aconselhável aumentar esse tamanho para 256 *bits*. A chave produzida deve ter o comprimento de 1024 *bits* [31].

Capítulo 3

Metodologia de Investigação e Ferramentas

O sistema desenvolvido e aqui proposto pretende, como já referido, permitir uma consulta segura de dados que têm de ser protegidos relativos a pacientes uma vez que é fundamental garantir a privacidade dos doentes.

Neste capítulo, apresenta-se a metodologia de investigação utilizada neste projeto, assim como as ferramentas e tecnologias usadas na sua implementação.

3.1 Metodologia de Investigação

Para planear e produzir o sistema em concreto, foi utilizada uma metodologia de investigação ligeiramente melhorada baseada no Modelo em Cascata. Este último é um dos modelos mais clássicos e mais usados em projetos de engenharia de *software*. Necessita de um grande esforço inicial na fase de planeamento o que permite que ocorram menos erros nas fases posteriores de implementação. O processo começa com a definição dos requisitos do sistema, dos requisitos de *software*, da arquitetura do sistema, dos componentes do sistema, todos em cadeia, de forma independente e pela ordem em que foram enumerados. Depois de todas estas fases completas, inicia-se o processo de implementação do código. Findo este processo, existe uma fase de

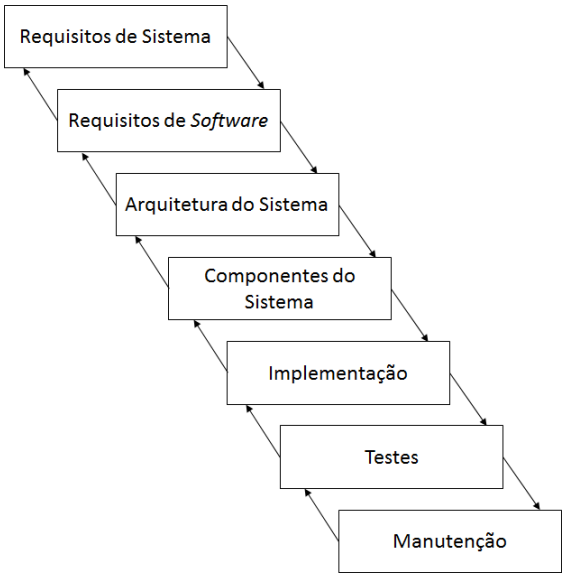


Figura 3.1: Esquema do Modelo em Cascata (adaptado de [2]).

testes e posterior manutenção do sistema em funcionamento no cliente. Este processo está esquematizado na figura 3.1. Nenhuma das fases se sobrepõe a outra em termos temporais. Apesar disso, é possível voltar atrás, se bem que isso irá implicar grandes mudanças nas fases seguintes [2].

As grandes vantagens deste modelo são a sua simples implementação e fornecer bons hábitos de desenvolvimento (planeamento antes da implementação). Pelo contrário, é inflexível. Apenas a fase final produz um produto que não é um documento e o retorno a uma fase anterior vai exigir grandes correções nas fases seguintes [2].

Tal como referido, este projeto baseou-se numa metodologia de investigação modificada, relativamente ao Modelo em Cascata, designado simplesmente por Modelo de Cascata Modificado. Este é mais flexível, permitindo que as diferentes fases se sobreponham na linha temporal de execução do projeto, possibilitando mais flexibilidade na implementação do projeto. Outras vantagens são, por exemplo, reduzir a documentação já que há uma ligação entre fases e nem tudo tem de ser escrito, podendo ser implementado diretamente e permitir que fases mais fáceis sejam executadas antes de algumas fases mais difíceis. Por outro lado, podem aparecer alguns erros e algumas

inconsistências porque pode não haver comunicação entre fases [2].

Este modelo é adequado ao projeto desta tese, visto que vai ser executado individualmente e não há falhas de comunicação entre as diferentes fases. Por outro lado, faculta ao programador definir bem a marioria dos requisitos e implementar algumas peças da aplicação enquanto termina a especificação dos restantes.

3.2 Ferramentas

3.2.1 Programação em Java e *NetBeans*

A linguagem de programação Java é uma linguagem de propósitos gerais, baseada em classes e orientada a objetos. É uma linguagem considerada de alto nível, ao fornecer ao programador uma gestão eficiente na alocação de memória e processo inverso, por exemplo [32].

Utilizando a mesma linguagem, recorreu-se ao Java Swing para criar a interface gráfica. O Java Swing é descrito como um Graphical User Interface (GUI) *toolkit*, permitindo criar interfaces gráficas para grandes aplicações de uma forma relativamente simples, tendo como apoio um grande número de componentes disponíveis altamente customizáveis em aspeto e comportamento [33].

Para facilitar a implementação das aplicações em Java, foi utilizado um Integrated Development Environment (IDE), neste caso, o *NetBeans*. Este garante editores para escrita do código e análise do mesmo. Ao mesmo tempo, possui janelas específicas para construção das interfaces com Java Swing de uma forma visual (gerando o código automaticamente). Outra das características que levou à escolha desta aplicação foi o suporte para *Maven* e ser *open source* [34].

O *Maven* é um projeto de gestão de *software* que permite gerir todo o código do projeto e documentação a partir de um ponto central. Neste caso, esta tecnologia foi utilizada para importar bibliotecas necessárias para a aplicação Java [35].

3.2.2 Componente *poreid*

A biblioteca *poreid* é um componente Java utilizado para permitir a integração do Cartão de Cidadão da República Portuguesa numa aplicação na referida linguagem. Esta biblioteca é de uso livre e permite executar todas as operativas necessárias sobre o cartão de cidadão nomeadamente operações criptográficas. O uso deste componente recorreu ao *Maven*, sendo que foi importado a versão 1.48 deste projeto através da seguinte configuração [36]:

```
<dependency>
  <groupId>org.poreid</groupId>
  <artifactId>poreid</artifactId>
  <version>1.48</version>
</dependency>
```

3.2.3 *Framework .NET*

A plataforma *.NET* é um componente integrado no Sistema Operativo *Microsoft Windows* para criar e manter em funcionamento aplicações de *software* e *Web Services*. De certa forma, é um modelo simplificado para programar e instalar aplicações no referido Sistema Operativo [37].

Esta plataforma é composta por quatro componentes:

- Um conjunto de diversas linguagens e tecnologias usadas para criar *Web Forms*, *Web Services* e aplicações para *Windows*.
- Um conjunto de ferramentas que permite desenvolver o código, compilá-lo e operá-lo.
- Uma série de servidores para correr *Web Services* e outras aplicações.
- Um núcleo de serviços que executa tarefas base que permitem que os programadores desenvolvam em cima das mesmas.

Esta tecnologia, apesar de ser bastante versátil e possuir inúmeras vantagens, foi usada, pois já existia um *Web Service* que fornecia algumas das

funções necessárias de conexão e obtenção de dados da Base de Dados e optou-se por utilizá-lo e acrescentar algumas funcionalidades ao mesmo.

A ligação à Base de Dados é permitida, aplicando o *Namespace System.Data.OracleClient* que garante uma ligação e acesso aos dados de uma Base de Dados Oracle.

3.2.4 Base de Dados *Oracle*

Uma base de dados é definida pelo Linux Information Project (LINFO) como um conjunto de dados com uma estrutura regular e organizada de uma forma computacional e facilmente acessível. Cada registo da mesma é composto por vários campos ou atributos. No caso deste projeto, cada registo corresponde a um relatório ou a um médico [38].

Existem diversos tipos de bases de dados, neste caso, utilizou-se uma Base de Dados *Oracle* inserida no tipo Relacional. Neste, os dados são guardados em tabelas relacionadas entre elas. Devido a estas relações, possuem elevada *performance* [38].

Este sistema controla a concorrência no acesso aos dados com eficiência mesmo para um elevado número de utilizadores; permite a leitura e modificação dos dados de forma consistente e proporciona um elevado desempenho, para possibilitar o máximo de produtividade [39].

SQL é a linguagem de programção utilizada para definir e manipular a Base de Dados *Oracle*. Esta gestão é feita através de uma aplicação da mesma empresa designada por *Oracle SQL Developer* [39].

A eleição de uma Base de Dados deste género e desta empresa deveu-se a ser aquela que neste momento está implementada no CHP, Hospital para o qual foi realizado o sistema aqui descrito.

3.2.5 *Framework Django*

O *Django* é um *framework* direcionado para o desenvolvimento de páginas *Web* baseado na linguagem de programação de alto nível *Python*. Permite poupar tempo na construção das aplicações e facilita a sua manutenção com o mínimo de esforço. Baseia-se num padrão de desenvolvimento designado

como MVC. Este padrão permite separar o modelo de dados e consequente definição e acesso aos mesmos (*model*) da lógica e direcionamento dos pedidos efetuados na página (*controller*). Este último está ao mesmo tempo separado da interface gráfica apresentada ao utilizador (*view*) [40, 41].

As grandes vantagens desta tecnologia são a quantidade reduzida de código que é necessário escrever, a eficiência (em termos temporais) com que se obtém resultados e a promoção da filosofia Don't Repeat Yourself (DRY) que favorece a reutilização de código para funções similares [40].

Outra funcionalidade chave que foi tida em conta, aquando da decisão de utilizar esta *Framework*, foi a capacidade de acesso a uma base de dados relacional *Oracle* e geração automática dos modelos a partir da mesma [42].

A facilidade de criação de uma página de administração da Base de Dados com adição, remoção e edição de registos e possibilidade de criação de diferentes usuários com vários níveis de acesso levou à escolha desta tecnologia para este projeto. A edição de todo o código foi levada a cabo, usando um simples editor de texto como o *Sublime Text 2*.

Capítulo 4

Análise de Requisitos de Segurança

O presente capítulo apresenta a análise de requisitos de segurança, fazendo também uma análise de requisitos do sistema. Os requisitos de segurança foram inspirados no *Common Criteria for Information Technology Security Evaluation*, sendo que, para facilitar a leitura deste documento, foram adaptados para uma notação própria mais simples e, sem dúvida, mais amigável para o leitor.

Para tal, foram usadas as três partes do documento: *Introduction and general model* [43], *Security functional requirements* [44] e *Security Assurance components* [45] e, ainda, um guia menos técnico [46].

Os requisitos do sistema e o *design* do sistema seguem algumas recomendações dadas por *Bourque, Pierre et al.* [47].

4.1 Descrição Geral

Um Registo Clínico Eletrónico é um repositório de dados de doentes em formato digital. Estes são armazenados e acedidos de forma segura e consultados, apenas, por utilizadores autorizados. Contem informação de diversos tipos (texto, imagens, etc) e a sua principal função é o suporte contínuo e eficiente do serviço de prestação de saúde [48].

4.1.1 Perspetiva de Produto

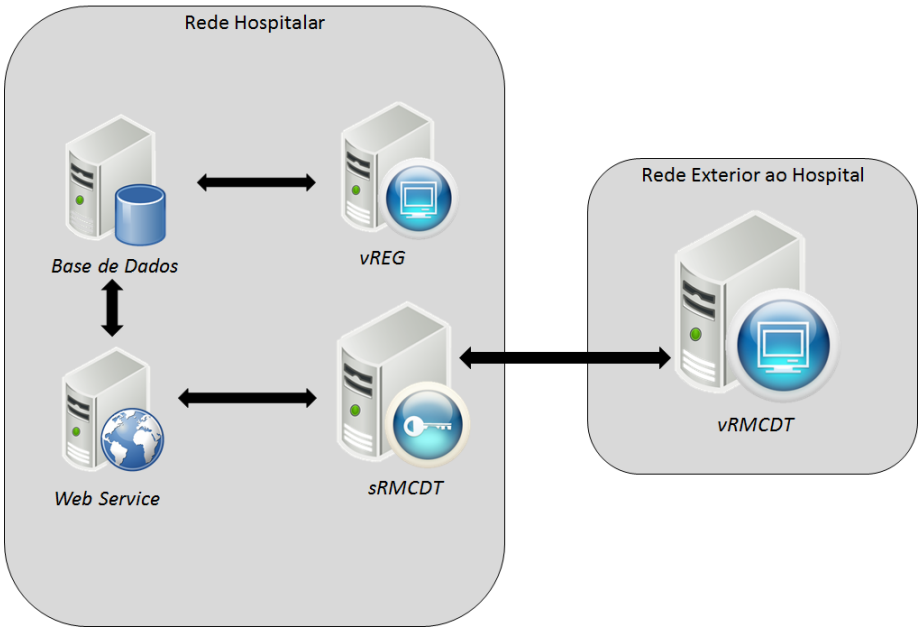


Figura 4.1: Esquema do Produto a Desenvolver Inserido no Ambiente em que Funcionará.

O Sistema Seguro de Acesso ao Registo Clínico Eletrónico (SSARCE) representado na figura 4.1 é um esquema que mostra o funcionamento geral do projeto a desenvolver. Este consiste em dois terminais: o Visualizador de Relatórios de Métodos Complementares de Diagnóstico e Terapêutica (vRMCDT) e o Visualizador de Registos (vREG). Haverá, ainda, um servidor designado por Servidor de Relatórios de Métodos Complementares de Diagnóstico e Terapêutica (sRMCDT) que atenderá os pedidos efetuados por um ou mais terminais vRMCDT. O terminal com interface vREG terá acesso direto à base de dados onde se guardam os registos. O *Web Service*, já está implementado e funcional, responde a pedidos de relatórios do sRMCDT que, por sua vez, os enviará para o terminal do vRMCDT que requisitou esse mesmo relatório. Será na mesma base de dados do CHP que se guardarão os dados relativos aos registos de acessos, aos médicos e administradores.

Existirão dois tipos de utilizadores do sistema, por um lado, o vRMCDT será usado por médicos, por outro, administradores poderão consultar o uso

do sistema através do vREG. Para facilitar o entendimento deste documento, ao longo de todas as imagens, os dois tipos de utilizadores serão representados pelos símbolos presentes na figura 4.2.

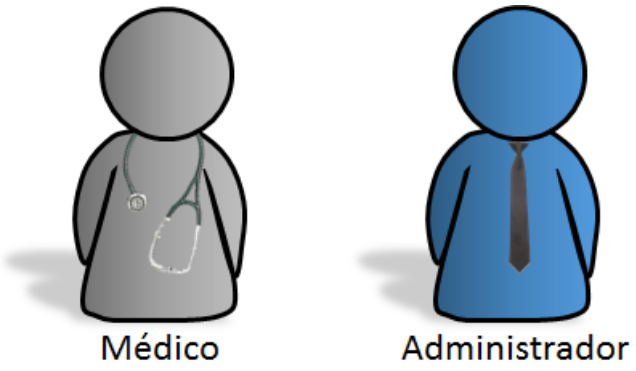


Figura 4.2: Símbolos que Representam os Possíveis Utilizadores do Sistema

4.1.2 Restrições Gerais

O vRMCDT irá ser utilizado em diversas máquinas por médicos que tenham interesse em visualizar os relatórios. Daí que esta aplicação deva ser a mais "leve" possível, para que consiga ser utilizada em computadores com características inferiores, caso necessário. Estes deverão, também, ter acesso à *Internet*.

O sRMCDT será instalado, obrigatoriamente, numa máquina com acesso à Intranet através da rede hospitalar. Esta máquina deverá, ainda, ser capaz de fazer pedidos de relatórios ao *Web Service*. A velocidade de consulta dos relatórios será limitada pela velocidade da rede, da máquina e número de pedidos a serem respondidos no momento. Para que possa haver acesso do exterior a este serviço, terá de haver uma exceção na *firewall* correspondente ao IP e porta onde estará ativo o processo relativo ao sRMCDT.

O vREG funcionará numa ou várias máquinas que estejam diretamente ligadas à Intranet do CHP e terá ligação direta à Base de Dados.

A implementação da aplicação está, também, limitada pela base de dados já que é necessário usar a existente (Base de Dados *Oracle*).

O vRMCDT e o sRMCDT devem comunicar pelo protocolo TCP/IP,

enquanto a comunicação entre o sRMCDT e o *Web Service* deverá seguir as normas do protocolo Simple Object Access Protocol (SOAP).

4.1.3 Características de Utilizadores

Para um correto funcionamento da aplicação, é necessário que um médico que a utilize esteja registado como um médico na base de dados.

Um administrador tem de estar registado na base de dados e aceder ao sistema através do seu *username/password*.

4.2 Análise de Requisitos

As subsecções seguintes enumeram as funções que será possível executar no SSARCE por ambos os utilizadores já referidos, médicos e administradores.

4.2.1 Visualização de um Relatório por um Utilizador Médico

Diagrama:

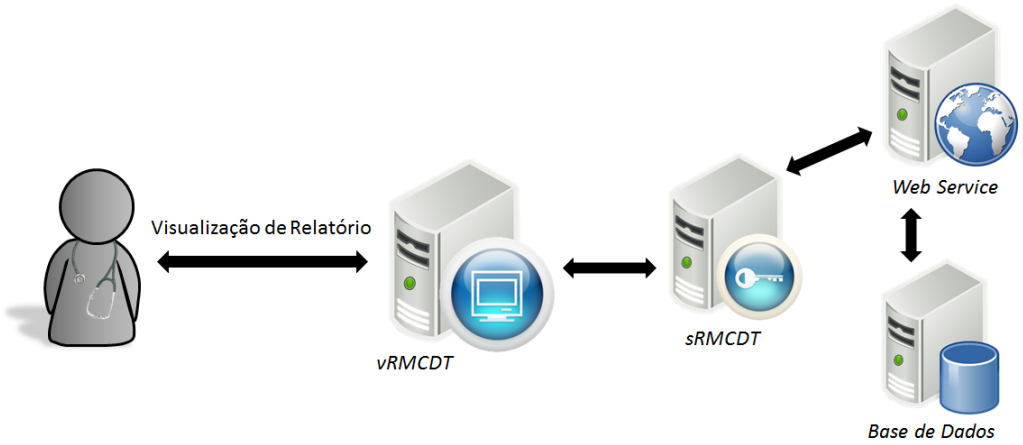


Figura 4.3: Diagrama Representativo da Visualização de um Relatório por um Médico.

Descrição breve: um médico conecta-se pelo terminal vRMCDT, pesquisa o relatório do exame que deseja consultar e visualiza-o.

Descrição passo a passo: Antes da visualização do relatório, o médico apenas iniciou a aplicação correspondente ao cliente (vRMCDT).

1. O médico pesquisa todos os relatórios dos exames/análises realizados pelo doente, através do seu nome, podendo filtrar a pesquisa, quanto à modalidade do exame realizado.

2. O servidor requisita a informação à base de dados, que, posteriormente, é enviada para o terminal, onde é mostrada ao profissional de saúde.

3. O médico escolhe o relatório que deseja visualizar, verificando, anteriormente, informações detalhadas do exame, como número do episódio hospitalar, o número do exame, a data do exame, o número e data do pedido, a descrição do exame, o local e o técnico que o efetuou.

4. O servidor requisita, então, o relatório em formato *pdf* ao *Web Service*, enviando-o, novamente, ao programa controlado pelo médico.

5. O médico consulta o relatório.

Pré-condições: O médico tem de estar autenticado no sistema.

Pós-condições: É necessário que haja registo da visualização do relatório por parte do utilizador.

4.2.2 Visualização de Registos por parte de um Utilizador Administrador

Diagrama:

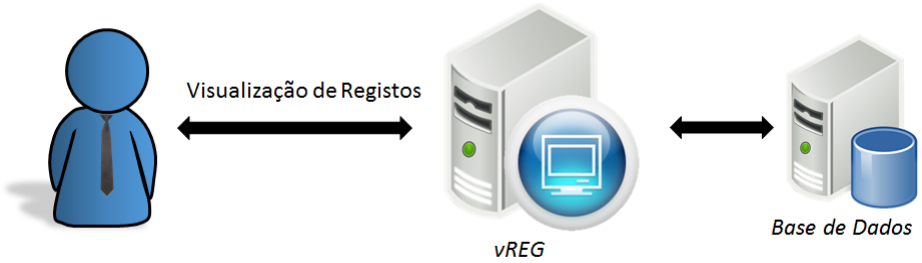


Figura 4.4: Diagrama Representativo da Visualização de Registos de Consultas de Relatórios por parte de um Administrador

Descrição breve: um administrador conecta-se pelo terminal vREG, pesquisa os registos que deseja consultar e visualiza-os.

Descrição passo a passo: Antes da visualização dos registos, o administrador apenas iniciou a aplicação correspondente ao vREG.

1. O administrador pesquisa os registos que deseja visualizar, através da data, tipo de exame, paciente ou médico que efetuou uma possível consulta.

2. O servidor requisita a informação à base de dados, que posteriormente é enviada para o terminal vREG, onde é mostrada ao utilizador.

3. O administrador consulta os registos.

Pré-condições: O administrador tem de estar autenticado no sistema.

4.2.3 Registo de um Utilizador Médico por um Administrador

Diagrama:

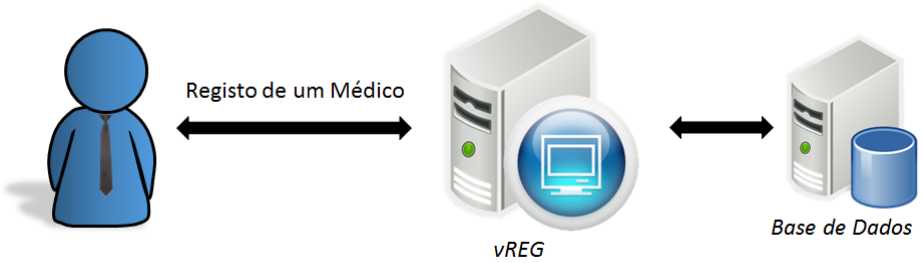


Figura 4.5: Diagrama Representativo do Registo de um Médico por parte de um Administrador

Descrição breve: um administrador conecta-se pelo terminal vREG, introduz as informações do médico que deseja registar e confirma as ações, "inscrevendo"o médico nos possíveis utilizadores do terminal vRMCDT.

Descrição passo a passo: Antes do registo do médico, o administrador apenas iniciou a aplicação correspondente ao vREG.

1. O administrador introduz os dados relativos ao médico que quer registar, sendo estes, número do Cartão de Cidadão e prazo máximo de acesso à aplicação por esse utilizador (médicos registados no hospital terão este campo nulo).

2. O mesmo utilizador visualiza de novo as informações e confirma-as, caso estejam certas, podendo voltar ao passo 1, para alterar/corrigir o que achar necessário.

3. A aplicação vREG introduz as mesmas informações na base de dados.

4. Esse médico fica apto a utilizar o vRMCDT.

Pré-condições: O administrador tem de estar autenticado no sistema.

4.2.4 Alteração de um Utilizador Médico por um Administrador

Diagrama:

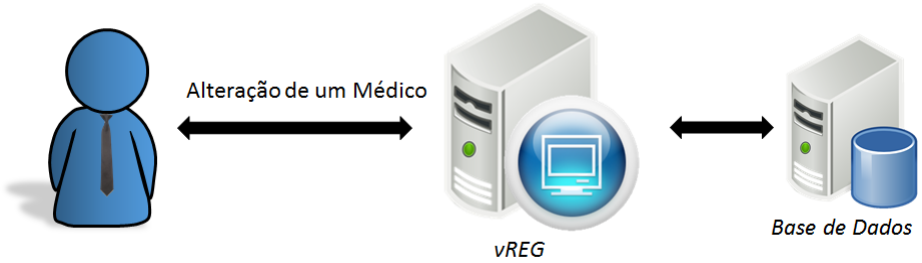


Figura 4.6: Diagrama Representativo da Alteração dos Dados de um Médico por parte de um Administrador

Descrição breve: um administrador conecta-se pelo terminal vREG, pesquisa o médico relativo ao qual quer fazer as alterações, escolhe o utilizador, modifica os dados e guarda as alterações efetuadas.

Descrição passo a passo: Antes do registo do médico, o administrador apenas iniciou a aplicação correspondente ao vREG.

1. O administrador pesquisa o médico relativo ao qual quer alterar os dados.

2. O mesmo utilizador altera os dados do médico que selecionou e guarda as alterações.

3. A aplicação vREG altera as informações na BD.

Pré-condições: O administrador tem de estar autenticado no sistema.

4.2.5 Eliminação de um Utilizador Médico por um Administrador

Diagrama:

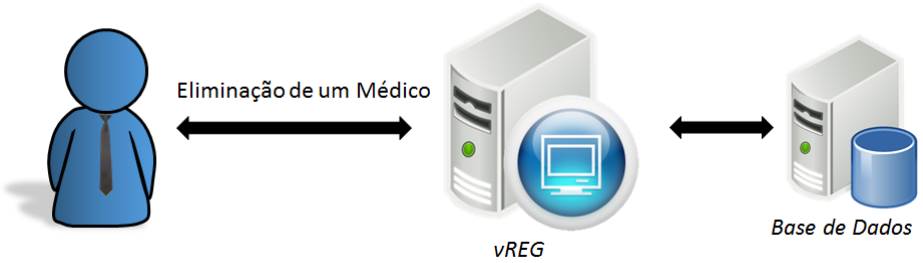


Figura 4.7: Diagrama Representativo da Eliminação dos Dados de um Médico por parte de um Administrador

Descrição breve: um administrador conecta-se pelo terminal vREG, pesquisa o médico que deseja apagar da BD e, dessa forma, retira-lhe o acesso à aplicação vRMCDT.

Descrição passo a passo: Antes da eliminação do médico, o administrador apenas iniciou a aplicação correspondente ao vREG.

1. O administrador pesquisa o médico relativo ao qual quer apagar os dados.

2. O mesmo utilizador escolhe o médico que deseja remover e confirma a ação.

3. A aplicação vREG apaga os dados da BD.

Pré-condições: O administrador tem de estar autenticado no sistema.

4.3 Requisitos de Interfaces Externas

As interfaces visualizadas nesta secção são meramente ilustrativas e servem para exemplificar como será o aspeto gráfico da aplicação final. Ao longo da implementação, poderá ser necessário efetuar algumas mudanças.

4.3.1 Interface da Aplicação vRMCDT

Ecrã Inicial

Figura 4.8: Ecrã Inicial da Aplicação vRMCDT

Esta janela (figura 4.8) será a primeira que aparece, ao iniciar a aplicação vRMCDT. Um médico conseguirá, apenas, avançar para uma janela diferente, caso se autentique corretamente perante o servidor sRMCDT. Esta autenticação será explicada mais à frente neste documento.

Caso falhe a autenticação, deverá aparecer uma mensagem de erro que indique ao utilizador qual o problema que ocorreu.

Ecrã de Pesquisa e Visualização de Relatórios

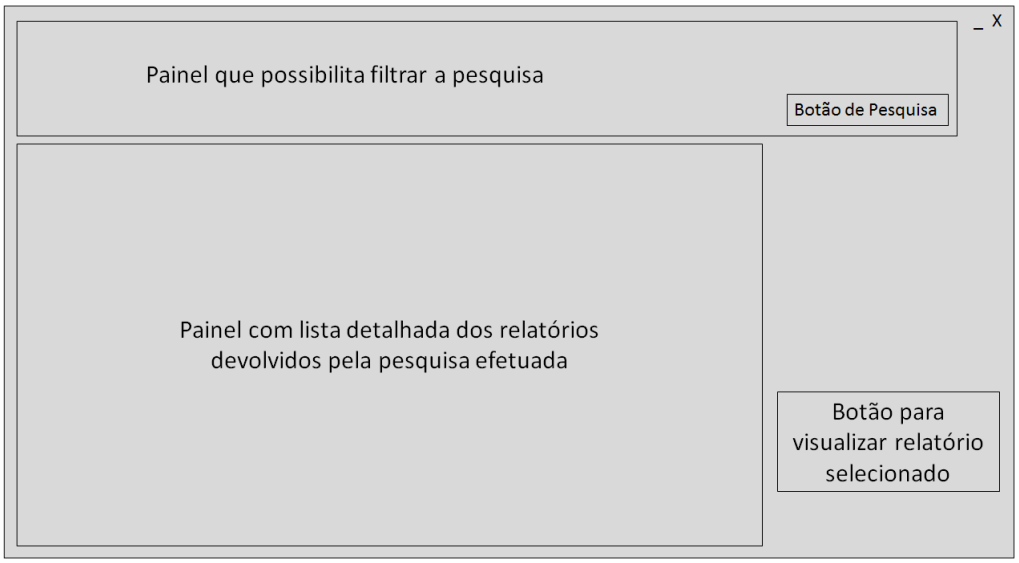


Figura 4.9: Ecrã de Pesquisa e Visualização de Relatórios vRMCDT

Nesta configuração (figura 4.9), o médico pode pesquisar por um relatório específico, filtrando a busca com diferentes campos que aparecerão no painel superior. Os resultados da pesquisa aparecerão no painel que se visualiza em baixo à esquerda em forma de lista, assim que se carregar no Botão de Pesquisa. Todos os elementos dessa lista são selecionáveis e estando um deles escolhido, ao carregar no botão visível à direita, abrir-se-á uma janela com o relatório. Será possível abrir mais do que um relatório ao mesmo tempo (em janelas diferentes) para análise da evolução do doente ou comparação com outros exames.

Caso haja alguma falha, esta deverá ser reportada ao utilizador com uma janela de erro que identifique o problema.

4.3.2 Interface da Aplicação vREG

Ecrã Inicial

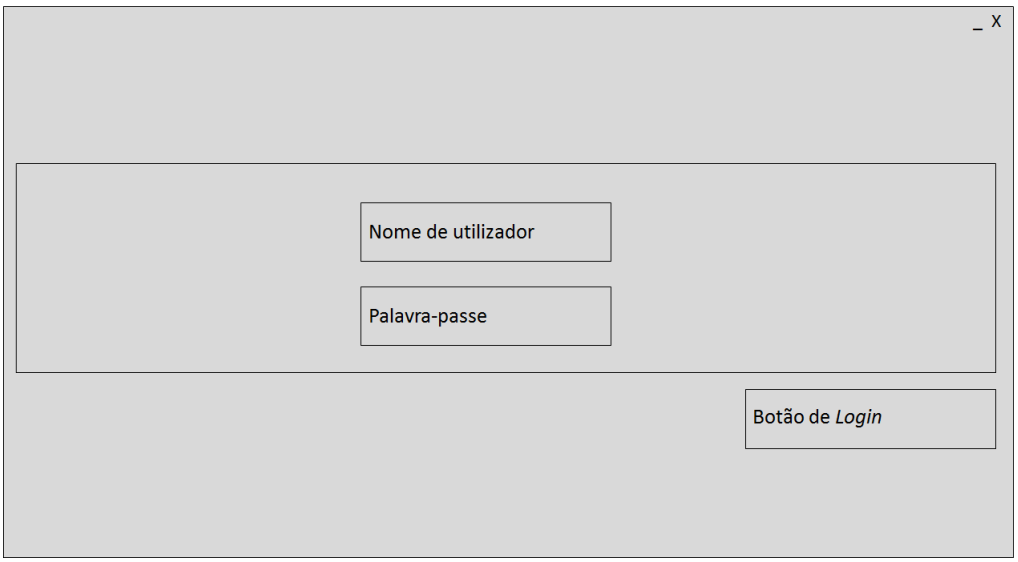


Figura 4.10: Ecrã Inicial da Aplicação vREG

Esta janela (figura 4.10) será a primeira que aparece, ao iniciar a aplicação vREG. Um administrador conseguirá, apenas, avançar para a janela seguinte, caso se autentique através do seu par nome de utilizador/palavra-passe. Esta autenticação será explicada mais à frente neste documento.

Ecrã de Pesquisa de Médicos

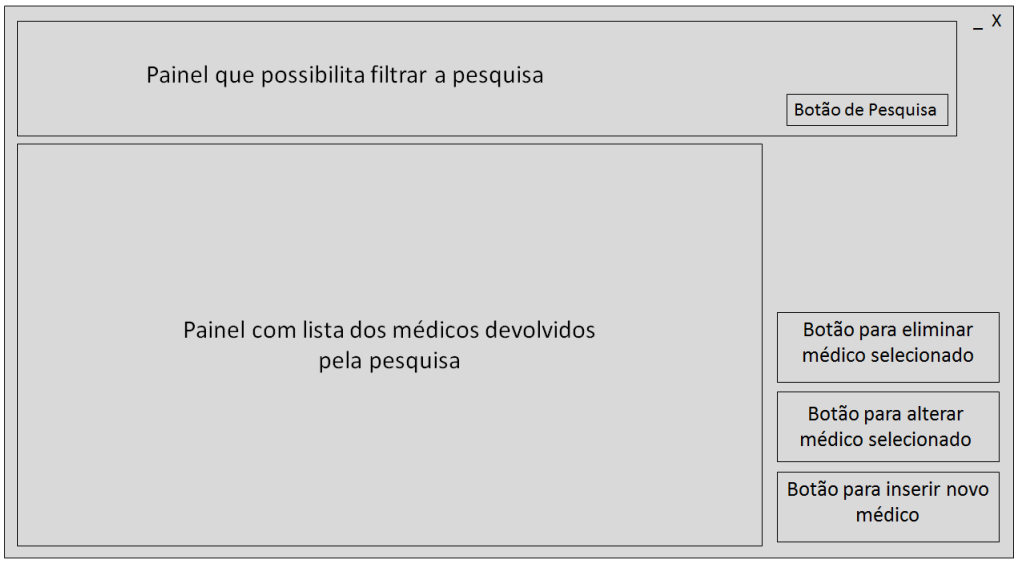


Figura 4.11: Ecrã para Pesquisa de Médicos da Aplicação vREG

No *design* da figura 4.11, o administrador pode pesquisar por um médico específico. Esta função funciona de forma similar à pesquisa de relatórios por parte de um médico, previamente analisado na secção anterior. Porém, o utilizador pode, agora, eliminar um médico ou alterá-lo. Se escolher a segunda opção, aparece a janela da figura 4.12, onde é possível alterar os dados e guardar as alterações.

Por outro lado, pode-se optar por criar um novo médico, carregando no Botão para inserir novo médico, aparecendo a janela da figura 4.13 que funciona de forma semelhante à janela para alterar os dados de um médico.

Todas as ações executadas por um administrador devem ser seguidas de uma janela com uma mensagem de sucesso da operação ou falha, apresentando nesse caso o motivo do erro.

Ecrã de Alteração de dados de um Médico

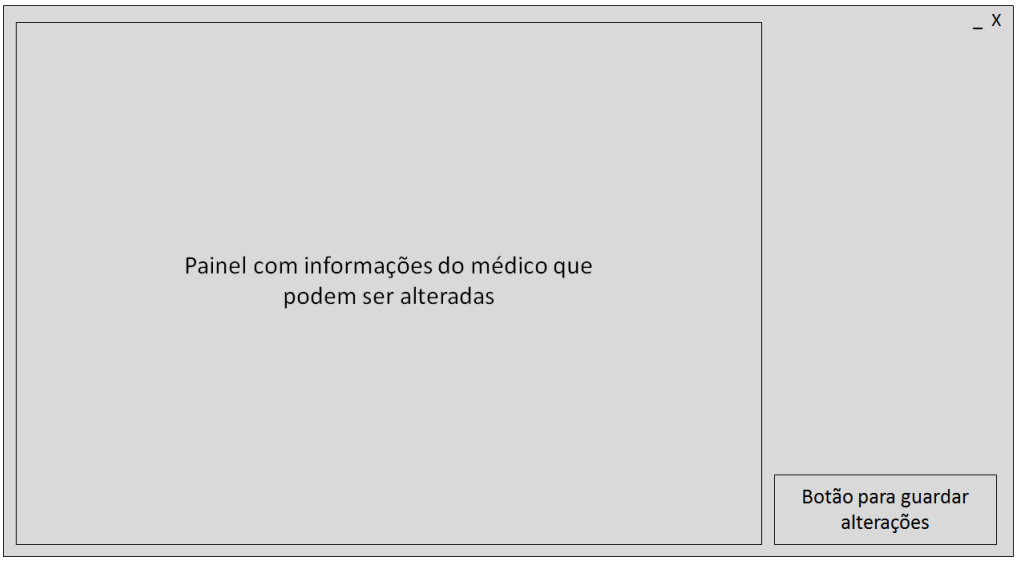


Figura 4.12: Ecrã para Alteração de Dados de Médicos da Aplicação vREG

Ecrã de Inserção de um novo Médico

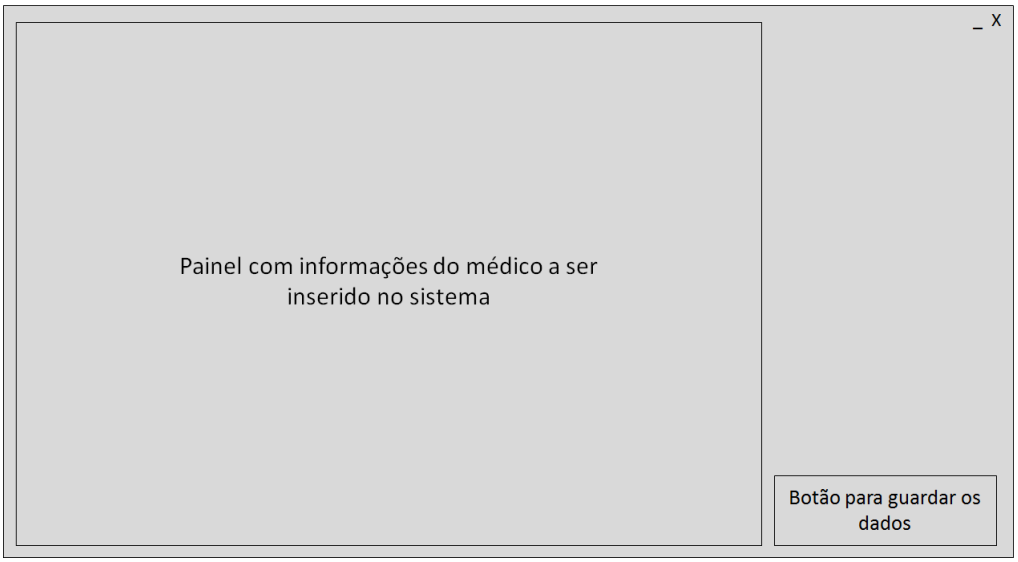


Figura 4.13: Ecrã para Inserir um Novo Médico na Aplicação vREG

4.4 Análise de Segurança

4.4.1 Definição do Problema de Segurança

Este capítulo tem como objetivo descrever quais as características do ambiente em que cada componente do SSARCE está inserido.

Assim, enumeram-se as suposições que são contempladas relativamente ao acesso físico do equipamento, ao grau de qualificação do pessoal, ao estado de funcionamento de cada máquina relativamente a *Malware* e aos canais de comunicação existentes.

Descrevem-se, ainda, as fontes e os ataques que podem ser executados sobre o sistema e que este deve solucionar.

Por fim, descrevem-se as políticas organizacionais que a aplicação deve ter em conta.

Suposições relativamente à Utilização e ao Ambiente

Antes da conceção do produto, são tidas em conta algumas suposições inicias que eliminarão certas preocupações, quanto à implementação do sistema.

1. S.1 Assume-se que apenas Administradores do Sistema têm acesso físico à infraestrutura de suporte da aplicação, nomeadamente, às máquinas em que estão em funcionamento a base de dados, o sRMCDT e o *Web Service*.

2. S.2 Declara-se que os Administradores são devidamente qualificados para gerir o sistema e que não irão fazer nenhuma alteração ao mesmo ou ter alguma ação considerada nociva para o seu funcionamento.

3. S.3 Acredita-se, também, que um utilizador com acesso válido ao vRMCDT (médico) não irá guardar ou divulgar informação sensível em qualquer situação.

4. S.4 Reconhece-se que as máquinas onde funciona cada componente do sistema não está afetada por nenhum tipo de *Malware*.

5. S.5 Admite-se que os canais de comunicação entre os componentes que funcionam na Intranet do CHP são completamente seguros e que não existe qualquer tipo de ataque sobre os pacotes que circulam nos mesmos.

Fonte de Riscos e Métodos de Ataque

Existem várias fontes de riscos para o funcionamento do sistema. Estas serão representadas pelo símbolo da figura 4.14 para um interpretação visual mais facilitada.

Figura 4.14: Símbolo que Representa um Indivíduo Considerado uma Fonte de Risco para o Funcionamento da Aplicação

Intruso com acesso a um terminal vRMCDT

Neste caso específico, um intruso desconhecido tem acesso a um terminal em que existe em funcionamento a aplicação vRMCDT, como se visualiza na figura 4.15. Isto pode ocorrer, por exemplo, quando um enfermeiro ou trabalhador do Centro de Saúde acede ao computador do mesmo ou no caso de um indivíduo ter conseguido obter a aplicação por outro qualquer meio.

Figura 4.15: Esquema Representativo do Acesso de um Intruso a um Terminal com o vRMCDT

Intruso com capacidade de visualizar/utilizar o Canal de Comunicação

Na situação representada na figura 4.16, o atacante consegue interferir com o canal de comunicação em diversas situações possíveis: servindo de intermediário à comunicação; vendo e gravando pacotes ou reenviando pacotes repetidos.

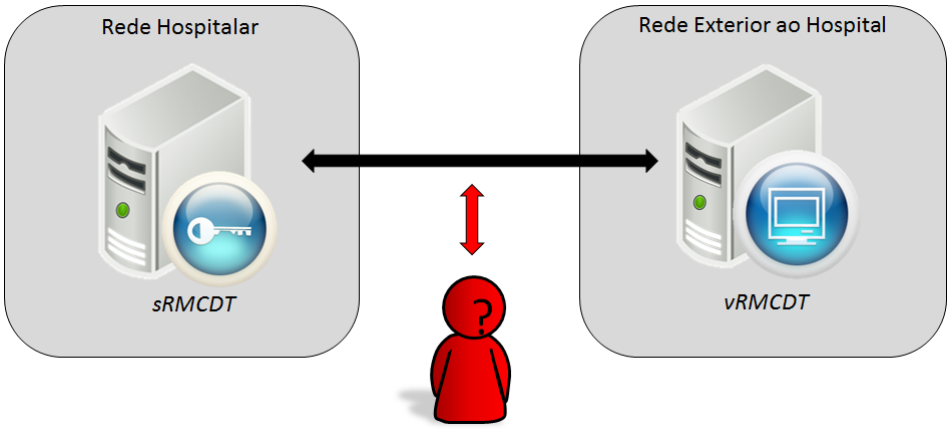


Figura 4.16: Esquema Representativo do Acesso de um Intruso ao Canal de Comunicação entre o vRMCDT e o sRMCDT

Possíveis Ataques à Segurança do Sistema

A tabela 4.1 ilustra os tipos de ataques, relacionando-os com o evento que ocorre no mesmo e as suas consequências nefastas no sistema.

Alvo do Ataque	Evento ocorrido	Descrição do evento	Consequências

Integridade	Modificação de informação	O atacante modifica os pacotes do canal de comunicação, enviando informações erradas ou sem sentido para os terminais	Perda de informação
	Spoofing	O atacante envia pacotes devidamente configurados para o servidor para que este envie informações específicas que o intruso deseja consultar	Divulgação de dados confidenciais
Confidencialidade	*Eavesdropping*	O atacante visualiza os pacotes que circulam no canal de informação, tentando obter alguma informação dos mesmos	Perda de informação e privacidade
	Divulgação de informação	O intruso com acesso válido a um terminal vRMCDT. Neste caso, divulga dados sensíveis e privados de doentes	Perda de privacidade e perda de confiança nos utilizadores do sistema
Negação de Serviço	Ataque por inundação	O atacante envia para o servidor um elevado número de pacotes, correspondendo a vários pedidos simultâneos	Serviço lento e corrompido

Autenticação	Personificação de um indivíduo legítimo	O atacante com acesso ao vRMCDT utiliza as credenciais de um médico válido, para aceder a informações confidenciais	Falsa representação de um utilizador e perda de confiança nos utilizadores do sistema
Autenticação	Personificação de um indivíduo legítimo	O atacante faz-se passar por um indivíduo legítimo sem acesso ao terminal vRMCDT, para obter informações privilegiadas	Falsa representação de um utilizador e crença que informações falsas são válidas

Tabela 4.1: Métodos de Ataque e as suas Consequências

Riscos Considerados na Posterior Análise de Segurança

Nos subcapítulos anteriores, foram apresentados diversos exemplos de possíveis ataques que violariam a segurança do sistema em questão. Esta secção faz uma síntese dos riscos que serão contornados através da implementação de características de segurança no sistema. Os mesmos são listados de seguida.

1. R.1 Um atacante recolhe os pacotes e tenta extrair informação dos mesmos

2. R.2 Um atacante modifica os pacotes, enviando informações erradas para um ou ambos os terminais

3. R.3 O intruso envia pacotes para o servidor, para tentar obter informações privilegiadas

4. R.4 O intruso faz-se passar por um indivíduo legítimo, para utilizar o serviço na totalidade

Políticas de Segurança da Organização

A Organização para a qual está a ser projetado este produto fez duas exigências devido às suas políticas internas. Estas restrições são listadas de seguida.

1. P.1 O sistema não deverá deixar a gravação dos relatórios no terminal vRMCDT, confiando-se que o utilizador final não irá utilizar outros meios para esse fim (fotografias ou a função do sistema de *Print Screen*).

2. P.2 Apenas utilizadores médicos registados na base de dados podem aceder à informação disponibilizada visualmente pelo terminal vRMCDT.

3. P.3 Só utilizadores designados como Administradores podem adicionar/remover médicos com acesso ao sistema.

4. P.4 Só utilizadores designados como Administradores podem aceder aos registos de visualizações de relatórios.

4.4.2 Objetivos de Segurança

É previsto que o SSARCE cumpra e satisfaça com eficiência alguns objetivos de segurança. Para os enumerar, optou-se por uma divisão em dois campos distintos. Inicialmente, definir-se-ão os objetivos para o sistema em si e, depois, para o ambiente em que se insere a aplicação.

Objetivos de Segurança do Sistema

Os seguintes objetivos referem-se ao funcionamento da aplicação e à forma como devem comunicar os terminais. Estes deverão ser garantidos aquando da implementação do SSARCE por parte do Programador do Sistema.

1. O.1 O sistema deve manter confidenciais todos os dados transmitidos entre o sRMCDT e o vRMCDT.

2. O.2 O sistema deve verificar a autenticidade de todas as mensagens trocadas entre o sRMCDT e o vRMCDT.

3. O.3 O sistema deve identificar todos os utlizadores do vRMCDT, antes de os deixar aceder a qualquer informação privada de doentes;

4. O.4 O sRMCDT deve restringir o acesso a utilizadores, de acordo com a política da organização P.2.

5. O.5 A aplicação vRMCDT deve autenticar o terminal sRMCDT, para garantir que comunica com um terminal seguro e confiável.

6. O.6 O terminal vREG deve autenticar os utlizadores, através do seu par: nome de utilizador/palavra-passe.

7. O.7 O termina vRMCDT não deve deixar o utilizador gravar os relatórios no sistema de ficheiros do seu terminal.

8. O.8 A aplicação deve garantir que todos os pedidos e envio de relatórios são registados.

Objetivos de Segurança do Ambiente Operacional do Sistema

Os seguintes objetivos dizem respeito ao ambiente em que funciona o SSARCE. Desta forma, apesar de serem objetivos quanto à segurança do sistema têm de ser garantidos por entidades exteriores à implementação da aplicação, sendo que, do ponto de vista do Programador, são pressupostos assegurados.

1. O.9 Os responsáveis pela manutenção da aplicação devem assegurar que todos os seus utilizadores são elucidados, quanto ao uso da mesma e quanto às consequências de uma utilização fraudulenta.

2. O.10 O ambiente operacional do sistema deve restringir o acesso físico às máquinas por parte de pessoal de manutenção não qualificado ou que não seja administrador do sistema.

3. O.11 O SSARCE será configurado de acordo com as normas e métodos definidos neste documento.

4. O.12 Os administradores não devem utilizar o seu poder para dar acesso à informação por parte de utilizadores que não cumpram as condições de acesso.

5. O.13 A comunicação entre os componentes que funcionam na Intranet do CHP deve ser segura.

6. O.14 O ambiente operacional deve ser livre de qualquer *Malware* que adultere o normal funcionamento do SSARCE.

Relação entre os Objetivos de Segurança e a Definição do Problema de Segurança

Ao longo deste subcapítulo, relacionam-se os objetivos de segurança com a definição do problema de segurança. De uma forma genérica, os riscos e as políticas da organização são acautelados pelos objetivos de segurança, sendo que as suposições tratam de parte desses objetivos. Por outro lado, alguns deles têm de ser cobertos por funcionalidades de segurança (levando ao subcapítulo 4.4.3), definidas sob a forma de requisitos funcionais de segurança.

Tabela 4.2: Relação dos Riscos com os Objetivos de Segurança

Riscos		Riscos e Objetivos de Segurança do Sistema	Objetivos do Ambiente do Sistema
R.1	Um atacante recolhe os pacotes e tenta extrair informação dos mesmos	O.1	
R.2	Um atacante modifica os pacotes, enviando informações erradas para um ou ambos os terminais	O.2	
R.3	O intruso envia pacotes para o servidor, para tentar obter informações privilegiadas	O.2; O.3; O.4; O.5	
R.4	O intruso faz-se passar por um indivíduo legítimo, para utilizar o serviço na totalidade	O.3; O.4; O.5	

Tabela 4.3: Relação das Políticas da Organização com os Objetivos de Segurança

Políticas de Segurança da Organização		Objetivos de Segurança do Sistema	Objetivos do Ambiente do Sistema
P.1	O sistema não deverá deixar a gravação dos relatórios no terminal vRMCDT, confiando-se que o utilizador final não irá utilizar outros meios para esse fim (fotografias ou a função do sistema de *Print Screen*)	O.7	O.9
P.2	Apenas utilizadores médicos registados na base de dados podem aceder à informação disponibilizada visualmente pelo terminal vRMCDT	O.2; O.3	O.12
P.3	Só utilizadores designados como Administradores podem adicionar/remover médicos com acesso ao sistema	O.6; O.11; O.12	
P.4	Só utilizadores designados como Administradores podem aceder aos registos de visualizações de relatórios	O.6; O.8	

Tabela 4.4: Relação das Suposições com os Objetivos de Segurança

Políticas de Segurança da Organização		Objetivos do Ambiente do Sistema
S.1	Assume-se que apenas Administradores do Sistema têm acesso físico à infraestrutura de suporte da aplicação, nomeadamente, às máquinas em que estão em funcionamento a base de dados, o sRMCDT e o *Web Service*	O.10
S.2	Declara-se que os Administradores são devidamente qualificados para gerir o sistema e que não irão fazer nenhuma alteração ao mesmo ou ter alguma ação considerada nociva para o seu funcionamento	O.12;
S.3	Acredita-se, também, que um utilizador com acesso válido ao vRMCDT (médico) não irá guardar ou divulgar informação sensível em qualquer situação	O.9
S.4	Reconhece-se que as máquinas onde funciona cada componente do sistema não está afetada por nenhum tipo de *Malware*	O.14
S.5	Admite-se que os canais de comunicação entre os componentes que funcionam na Intranet do CHP são completamente seguros e que não existe qualquer tipo de ataque sobre os pacotes que circulam nos mesmos	O.13

Conclusão acerca dos Objetivos de Segurança

Baseado nos objetivos de segurança e na sua relação com os riscos, suposições e políticas pode-se concluir que o problema de segurança definido no Capítulo de Definição do Problema de Segurança foi resolvido, já que os riscos foram contrariados, as políticas de segurança foram asseguradas e as suposições foram preservadas.

4.4.3 Requisitos Funcionais de Segurança

Este capítulo define os requisitos funcionais de segurança, para o sistema a desenvolver. Como ponto de partida, foi tido em conta o Common Criteria Part II [44], mas é utilizada uma notação própria para cada requisito, visto que os originais são muito exaustivos para o sistema relativamente simples que se está a desenvolver. Ainda assim, este padrão não foi abandonado, sendo que foi utilizado para que os requisitos definidos cubram tudo o que seria expectável.

1. F.1 O sistema deve gerar registos de visualização de relatórios de métodos complementares de diagnóstico e terapêutica.

2. F.2 Em cada registo definido anteriormente devem ser guardadas as seguintes informações:

   (a) Identificação de utilizador
   (b) Número do processo do paciente
   (c) Número do exame
   (d) Identificação do pdf
   (e) Data de acesso à informação
   (f) IP usado na altura da pesquisa

3. F.3 O sistema deve fornecer a utilizadores Administradores a capacidade de consultar os registos de visualização de relatórios de uma forma em que seja fácil a interpretação dos mesmo.

4. F.4 O sistema deve permitir a capacidade de pesquisar, ordenar e restringir a pesquisa de registos quanto a:

(a) Identificação de utilizador
(b) Número do Processo do paciente
(c) Número do Exame
(d) Identificação do *pdf*
(e) Data de acesso à informação
(f) IP usado na altura da pesquisa

5. F.5 O sistema não deve fornecer uma função para apagar registos, nem mesmo a Administradores.

6. F.6 O SSARCE deve ser capaz de acordar uma chave simétrica de 128 *bits* entre o vRMCDT e o sRMCDT, de acordo com o algoritmo Diffie-Hellman. O parâmetro p deve ter pelo menos 3072 *bits* e o parâmetro q deve ter, pelo menos, 256 *bits*.

7. F.7 O sistema deve ser capaz de cifrar dados com uma chave privada de um certificado do utilizador ou máquina, de acordo com o algoritmo RSA e tamanho de chave 4096 *bits*.

8. F.8 Ambos os terminais devem ser capazes de cifrar dados com a chave simétrica acordada com o algoritmo AES, sendo que a chave deve ter, pelo menos, 128 *bits*.

9. F.9 Para autenticar as mensagens trocadas, os sistemas devem conseguir executar o algoritmo HMAC-SHA2.

10. F.10 Através de certificados de Chave Pública contidos no Cartão de Cidadão, o sistema deve identificar inequivocamente os utilizadores, verificando se estes estão registando na Base de Dados como médicos.

11. F.11 Através de nome de utilizador/palavra-passe, o SSARCE deve autenticar utilizadores administradores.

12. F.12 O sistema deve avisar o utilizador, caso haja um erro de falta de integridade dos dados recebidos.

13. F.13 Os utilizadores devem ser identificados, antes de poderem executar qualquer operação ou visualizar quaisquer dados.

14. F.14 Através de um certificado de chave pública, o vRMCDT deve autenticar o servidor com que comunica, para garantir que é fidedigno.

15. F.15 O sistema não deve implementar uma função, para gravar os .pdfs na máquina do terminal vRMCDT.

Tabela 4.5: Relação dos Objetivos de Segurança do Sistema com os Requisitos Funcionais de Segurança

Objetivos de segurança do sistema		Requisitos Funcionais de Segurança
O.1	O sistema deve manter confidenciais todos os dados transmitidos entre o sRMCDT e o vRMCDT	F.6; F.7; F.8
O.2	O sistema deve verificar a autenticidade de todas as mensagens trocadas entre o sRMCDT e o vRMCDT	F.9; F.13
O.3	O sistema deve identificar todos os utlizadores do vRMCDT, antes de os deixar aceder a qualquer informação privada de doentes	F.10; F.13
O.4	O sRMCDT deve restringir o acesso a utilizadores, de acordo com a política da organização P.2	F.11
O.5	A aplicação vRMCDT deve autenticar o terminal sRMCDT, para garantir que comunica com um terminal seguro e confiável	F.14
O.6	O terminal vREG deve autenticar os utilizadores, através do seu par: nome de utilizador/palavra-passe	F.11
O.7	O termina vRMCDT não deve deixar o utilizador gravar os relatórios no sistema de ficheiros do seu terminal	F.15
O.8	A aplicação deve garantir que todos os pedidos e envio de relatórios são registados	F.1; F.2; F.3; F.4

4.4.4 Requisitos de Garantia de Segurança

Esta secção descreve os requisitos de garantia de segurança selecionados a partir dos requisitos de um nível EAL4, do ponto de vista do programador do sistema. Mais uma vez, tal como aconteceu com o capítulo anterior, será usada uma notação própria inspirada naquela descrita no Common Criteria Part III [45].

Classe de Garantia	Componente de Garantia		Método de Garantia do Requisito
Avaliação do Alvo de Segurança	G.1	Introdução ao Alvo de Segurança	Este documento está devidamente estruturado e descreve toda a aplicação a desenvolver.
	G.2	Definição do Problema de Segurança	Foi definido com clareza o problema de segurança que se impõe resolver.
	G.3	Objetivos de Segurança	Foram tidos em conta suposições, riscos e políticas, definindo-se e relacionando-se com os objetivos de segurança.
	G.4	Requisitos Funcionais de Segurança	Foram fornecidos todos os requisitos de segurança necessários, para resolver o problema definido inicialmente.

	G.5	Resumo de Especificações do Alvo de Segurança	Foi fornecida uma descrição em texto de como o alvo de segurança está de acordo com os requisitos funcionais de segurança.
Desenvolvimento	G.6	Descrição de Arquitetura de Segurança	O sistema foi pensado de maneira a corresponder a todos os requisitos funcionais de segurança.
	G.7	*Design* modular e básico	O sistema vai ser desenvolvido e pensado de forma a que todos os componentes internos sejam simples, justificando-se o porquê de cada um.
Guia de Operação	G.8	Procedimentos Prévios ao Uso	Os procedimentos para instalar o sistema e pô-lo completamente funcional vão ser assumidos pelo programador.
Testes e Verificações	G.9	Análise de Cobertura de Segurança	Todos os componentes vão ser testados e, da forma que foram pensados, prova-se que cobrem os requisitos de segurança teoricamente.

G.10	Teste e Verificação Funcional	O SSARCE vai ser testado e os resultados dos testes serão fornecidos.
G.11	Teste Independente	A aplicação poderá ser fornecida para teste por outras entidades.

Tabela 4.6: Requisitos de Garantia de Segurança

Todas as validações anteriormente apresentadas que levam à garantia de segurança não foram validadas por terceiros ou uma entidade externa ao CHP, mas é imperioso que isso seja executado.

4.5 Mecanismos de Segurança

Os requisitos funcionais de segurança são implementados e garantidos pelo Programador. É de notar que os requisitos com os números entre seis e dez, doze e catorze estão diretamente ligados a funções criptográficas, para poderem ser respeitados.

Desta forma, a comunicação entre o vRMCDT e o sRMCDT tem de ser implementada, usando mecanismos de segurança. Neste caso, fez-se uso do protocolo TLS que, tal como explicado no capítulo 1.1.8, permite criar um canal seguro com confidencialidade e autenticação de mensagens. A configuração utilizada tem de estar de acordo com os parâmetros definidos no capítulo 2.4, garantindo a atualidade do sistema.

Para isso, é necessário que o sRMCDT tenha um certificado próprio e que cada máquina em que está instalado o vRMCDT também tenha um (esta política está explicada detalhadamente no capítulo 4.7). Cada médico tem, também, um certificado próprio presente no seu cartão de cidadão.

Ao iniciar a comunicação com o TLS, as máquinas autenticam-se através dos seus certificados, criando um canal de comunicação seguro com as referidas características.

Posteriormente, é enviado um valor único ao vRMCDT que é assinado com a chave privada do certificado do médico. Esta assinatura é devolvida ao sRMCDT que a verifica, avaliando também as permissões desse utilizador quanto ao acesso à informação confidencial, autorizando ou não esse médico. Este processo (Role-based Access Control (RBAC)) é realizado através de informações guardadas na base de dados do sistema que determinam o cargo/responsabilidade (*role*) desse indivíduo perante o SSARCE.

Em suma, os referidos requisitos funcionais de segurança, que necessitam diretamente de funções criptográficas para a sua verificação, são efetivados, usando este mecanismo. Os restantes são postos em prática através da forma como é desenvolvido o SSARCE.

4.6 Arquitetura do SSARCE

Esta secção tem como objetivo explicitar o modo como cada terminal (vRMCDT, sRMCDT e vREG) está dividido em componentes lógicos que podem ser individualmente alterados, para correção de erros ou atualização, sem que isso afete o funcionamento do sistema.

É de notar que, tanto no vRMCDT quanto no sRMCDT, vários dos componentes poderiam ser unidos num só designado TLS. Optou-se por separá-los, para demonstrar as diferentes partes desse protocolo que se podem configurar, sendo, também, mais percetível os diferentes procedimentos que são executados durante a comunicação.

4.6.1 Arquitetura do vRMCDT

A aplicação vRMCDT será implementada em *Java*. Os componentes que a compõem podem ser visualizados na figura 4.17 e serão, posteriormente, explicados.

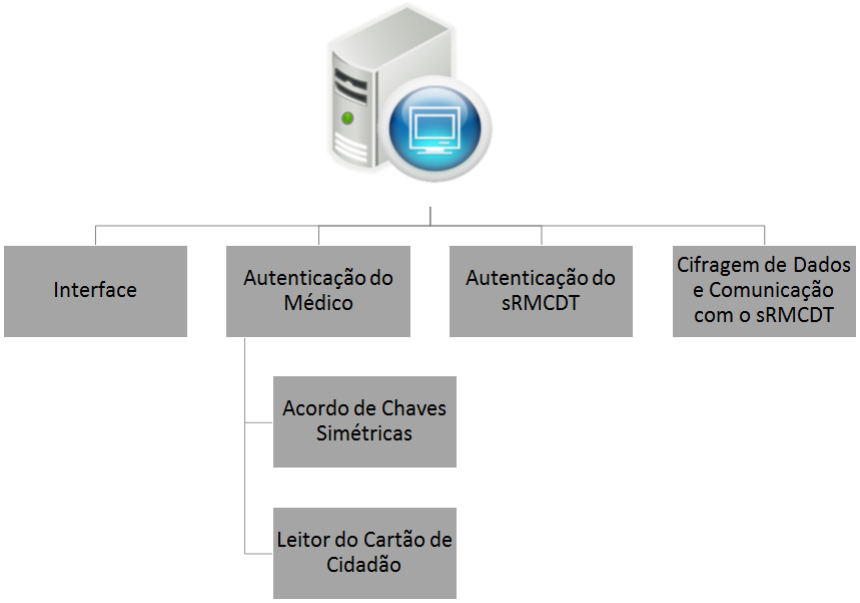


Figura 4.17: Componentes do vRMCDT

O componente Interface é aquele em que será implementado tudo o que será mostrado visualmente ao médico. É iniciado, ao correr a aplicação no respetivo terminal.

O componente Autenticação do Médico está subdividido em dois subcomponentes: o Acordo de Chaves Simétricas e Leitor de Cartão de Cidadão.

Ao iniciar o programa, o componente Acordo de Chaves Simétricas combina um chave simétrica com o sRMCDT. Esta chave é gerada através do algoritmo *Diffie-Hellman*. De seguida, o Leitor de Cartão de Cidadão recolhe dados do Cartão de Cidadão e assina um valor único enviado pelo sRMCDT com a chave privada contida no Certificado X.509, enviando a informação para o Servidor, devidamente cifrada com a chave simétrica e acompanhada de um MAC (processo a cargo do componente Cifragem de Dados e Comunicação com o sRMCDT). Desta forma, o sRMCDT pode verificar que o utilizador é fiável e identificá-lo inequivocamente. O componente Autenticação do sRMCDT analisa a cifra que lhe é fornecida, recorrendo à chave pública do Certificado do sRMCDT, para comprovar a autenticidade do terminal.

Por fim, a Cifragem de Dados e Comunicação com o sRMCDT gere toda a informação que é recebida do servidor, verificando a autenticidade das

mensagens. Ao mesmo tempo, cifra os dados a serem enviados com a chave simétrica combinada e verificada através do algoritmo AES e acompanhando a mensagem com um MAC.

Para que o componente Leitor do Cartão de Cidadão opere conforme expectável, é necessário ligar o Cartão de Cidadão a um leitor de cartões similar ao da figura 4.18 que se liga à máquina por USB.

Figura 4.18: Leitor de Cartões Compatível com o Cartão de Cidadão

4.6.2 Arquitetura do sRMCDT

A aplicação sRMCDT também será implementada em *Java*. Os componentes que a formam podem ser visualizados na figura 4.19 e serão posteriormente explicados.

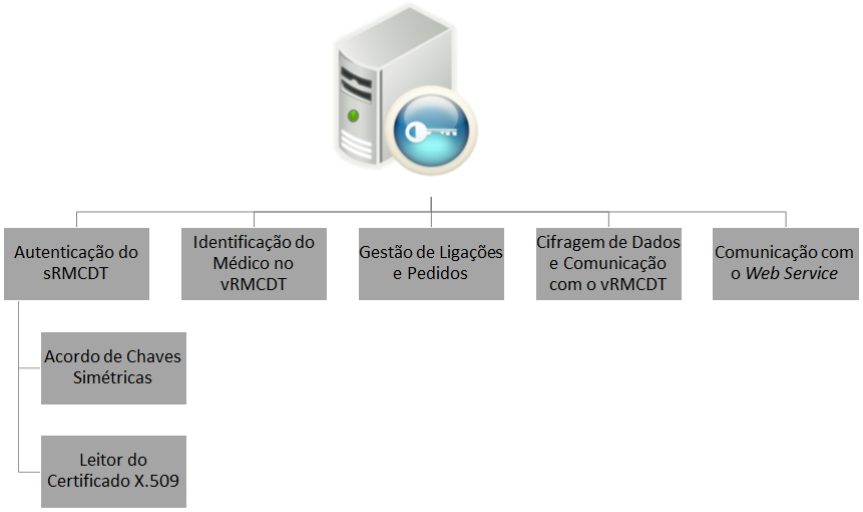


Figura 4.19: Componentes do sRMCDT

O componente Gestão de Ligações e Pedidos coordena os acessos de cada terminal vRMCDT, tratando de criar uma *thread* para cada pedido e guardar as informações relevantes de cada ligação.

O componente Autenticação do sRMCDT funciona de maneira análoga ao componente "Autenticação do Médico"no terminal vRMCDT. Implementa o acordo de chaves e verifica essa mesma chave, mas, desta vez, utilizando a chave privada do seu próprio certificado X.509. Da mesma forma, o componente Identificação do Médico no vRMCDT atua de forma idêntica à Autenticação do sRMCDT, para verificar a identidade do médico do outro lado da ligação. O mesmo se passa com o componente Cifragem de Dados e Comunicação com o vRMCDT.

Para pedir informações da Base de Dados e guardar os registos de acessos, existe o componente Comunicação com o *Web Service* que utiliza o protocolo SOAP, já anteriormente referido.

4.6.3 Arquitetura do vREG

A aplicação vREG também será implementada em *Java*. Os componentes que a constituem podem ser visualizados na figura 4.20 e serão posteriormente explicados.

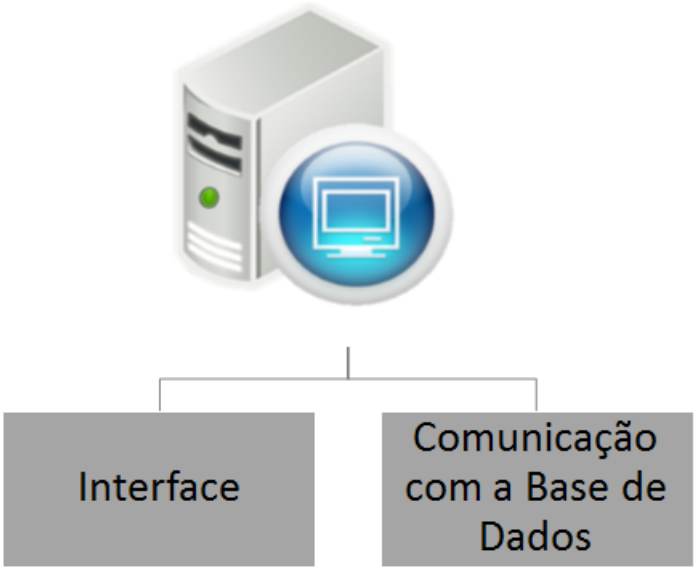


Figura 4.20: Componentes do vREG

O vREG é bastante simples e possui apenas o componente da interface gráfica (Interface) e o Comunicação com a Base de Dados que utiliza a API JDBC do *Java* para aceder à Base de Dados *Oracle* do CHP.

4.6.4 Comunicação entre Terminais do SSARCE

Visto o SSARCE ser um sistema distribuído, faz uso de protocolos de comunicação, para integrar todos os componentes. Estes já foram especificados ou referidos neste documento e são, agora, enumerados em conjunto. Os mesmos podem ser visualizados na figura 4.21.

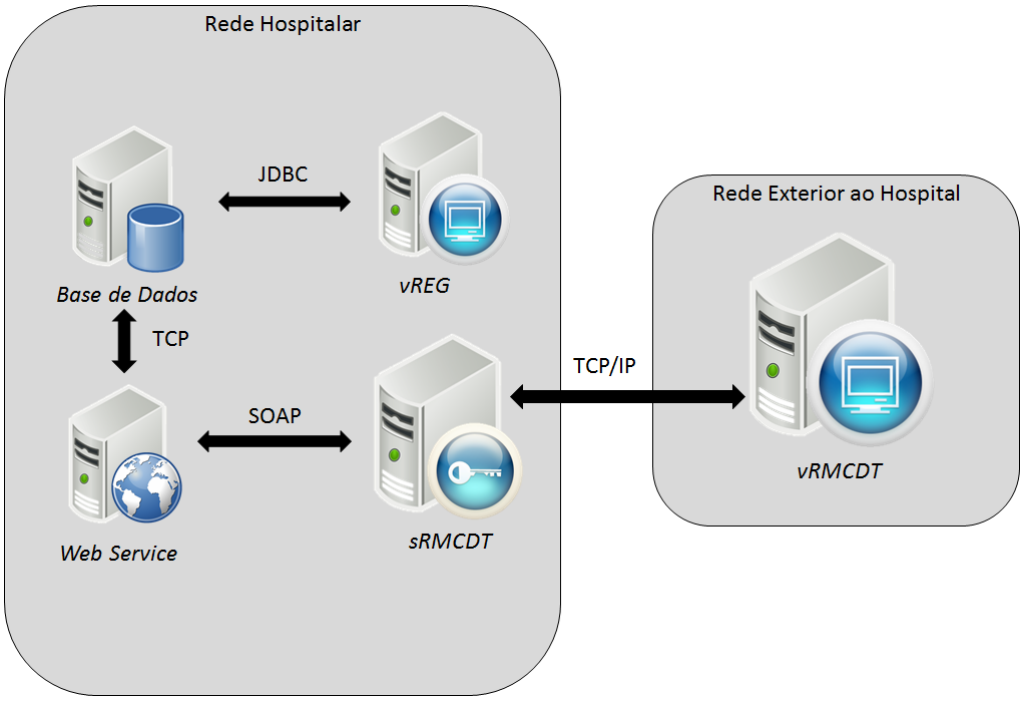


Figura 4.21: Protocolos de Comunicação no SSARCE

Entre o sRMCDT e o vRMCDT é utilizado o protocolo TCP/IP. Entre o sRMCDT e o *Web Service* é usado o protocolo SOAP. O vREG faz uso do JDBC, para aceder diretamente à Base de Dados *Oracle*. O *Web Service* liga-se à mesma Base de Dados, através do protocolo Transmission Control Protol (TCP).

4.7 Política de Certificados do SSARCE

Para o correto funcionamento do Sistema, foi criada uma política de certificados para garantir máxima segurança na comunicação entre os seus terminais.

Para além da política de certificados referida no capítulo 1.1.10 referente ao Cartão de Cidadão Português, foi implementada uma política para o próprio SSARCE com um propósito explicado mais à frente.

O certificado raiz do sistema tem as seguintes características:

- Nome Comum: SSARCE

- Unidade Organizacional: IT
- Organização: CHP
- Local: Porto
- País: Portugal
- Valido até: segunda-feira, 8 de fevereiro de 2016 09:56:45
- Emissor: o próprio
- Chave pública: do tipo RSA de 2048 bits

Por sua vez, um dos certificados que são assinados pela raiz da entidade emissora de certificados do SSARCE tem as seguintes características (sendo os restantes equivalentes, incrementado o número no nome comum):

- Nome Comum: vrmcdt1
- Unidade Organizacional: IT
- Organização: CHP
- Local: Porto
- País: Portugal
- Valido até: sexta-feira, 2 de setembro de 2016 11:02:59
- Emissor: SSARCE
- Chave pública: do tipo RSA de 2048 bits

Por outro lado, existe um único certificado assinado pelo SSARCE que não possui nenhum equivalente. Este certificado destina-se a ser usado pelo sRMCDT e tem os seguintes atributos:

- Nome Comum: srmcdt
- Unidade Organizacional: IT

- Organização: CHP
- Local: Porto
- País: Portugal
- Valido até: sexta-feira, 2 de setembro de 2016 11:06:35
- Emissor: SSARCE
- Chave pública: do tipo RSA de 2048 bits

A finalidade destes certificados é que cada máquina que tenha instalado o vRMCDT seja autenticada, criando mais uma verificação de segurança durante o processo e impedindo que máquinas não autorizadas repliquem o programa.

Conclusão, para que tudo funcione como esperado, é necessário que:

- O vRMCDT tem de confiar no certificado raiz do SSARCE
- O vRMCDT tem de ter o seu próprio certificado da máquina e conhecer a chave privada relativa a esse certificado
- O sRMCDT tem de confiar no certificado raiz do SSARCE
- o sRMCDT tem de ter o seu próprio certificado e conhecer a chave privada relativa a esse certificado
- O sRMCDT tem de confiar nos certificados que assinam os certificados de autenticação do cartão de cidadão

Esta política, definida especificamente para este projeto, pode, perfeitamente, ser substituída por um sistema comercial desde que se cumpram todos os requisitos de segurança anteriormente definidos (tamanhos de chaves, algoritmos, etc).

Capítulo 5

Resultados

O sistema desenvolvido durante este projeto, com o nome SSARCE, permite a consulta de relatórios de uma forma segura numa rede fora do ambiente hospitalar. Como também já foi referido, esta aplicação foi desenvolvida especificamente para o CHP, sendo que se destina a médicos que trabalhem em Centros de Saúde e necessitem de aceder a exames que tenham sido realizados no Hospital (vRMCDT). Foi também desenvolvida uma aplicação de administração (vREG) que se destina a pessoal administrativo e permite uma gestão muito simples de todo o sistema.

Ao longo deste capítulo, apresentam-se as duas aplicações que possuem interfaces gráficas, sendo que a arquitetura do sistema já foi apresentada no capítulo 4.

Posteriormente, é realizada uma análise SWOT a todo o sistema desenvolvido de forma a identificar os pontos fortes e fracos do mesmo. Isto permitirá não só enumerar os objetivos cumpridos como também os pontos a melhorar ou que necessitem de atenção imediata.

5.1 Aplicação vRMCDT

A aplicação vRMCDT aqui descrita foi desenvolvida de acordo com o planeamento apresentado no capítulo 4. Dessa forma, as suas janelas já tinham sido esquematizadas, sendo que agora as que se apresentam são as

que compõem a aplicação final.

Ao entrar na aplicação, o médico depara-se com uma janela de boas-vindas, que, ao mesmo tempo, informa o médico que deve colocar o seu Cartão de Cidadão num leitor próprio e conectá-lo ao seu computador. Esta janela pode ser visualizada na figura 5.1.

Figura 5.1: Janela de Boas-vindas da Aplicação vRMCDT

Ao carregar no botão Continuar, a aplicação lê os dados do Cartão de Cidadão, apresentando ao utilizador os seus dados como na figura 5.2 ou uma mensagem de erro como na figura 5.3.

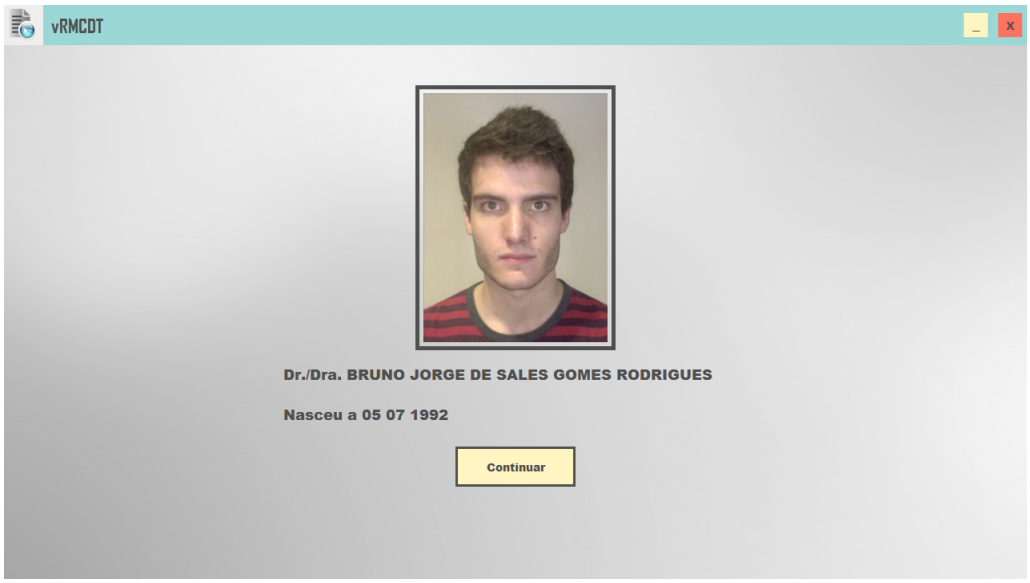


Figura 5.2: Janela de Apresentação de Dados do Médico

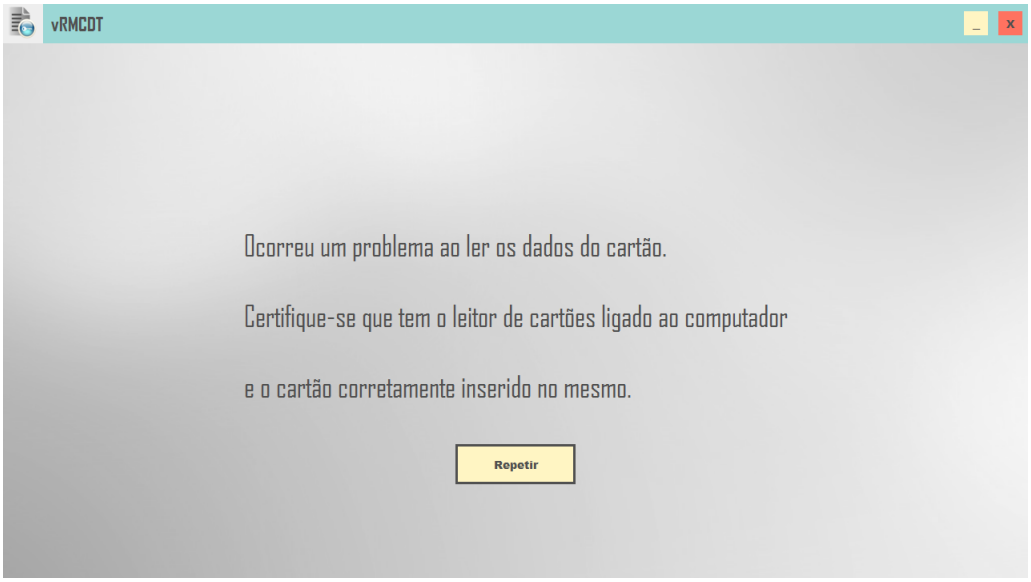


Figura 5.3: Janela de Erro, caso não seja possível ler o Cartão de Cidadão

Caso tenham sido lidos os dados do cartão corretamente, o médico ao carregar no botão Continuar vai iniciar a conexão ao sRMCDT. Durante esta fase, é pedido ao profissional de saúde que insira o seu PIN de autenticação relativo ao certificado de autenticação presente no seu cartão de cidadão

(5.4). O processo já descrito no capítulo 4 verifica se o médico possui autorização para aceder aos dados da aplicação. Em caso afirmativo é apresentada a janela da figura 5.5 ao utilizador. Nesta, o médico pode pesquisar o utente do qual deseja ver o(s) exame(s), podendo optar por uma pesquisa por nome ou número de episódio e ordenar os dados alfabeticamente (ordem ascendente ou descendente) ou por idade do doente (ordem ascendente ou descendente), visto que podem ser devolvidos mais do que um registo durante a pesquisa.

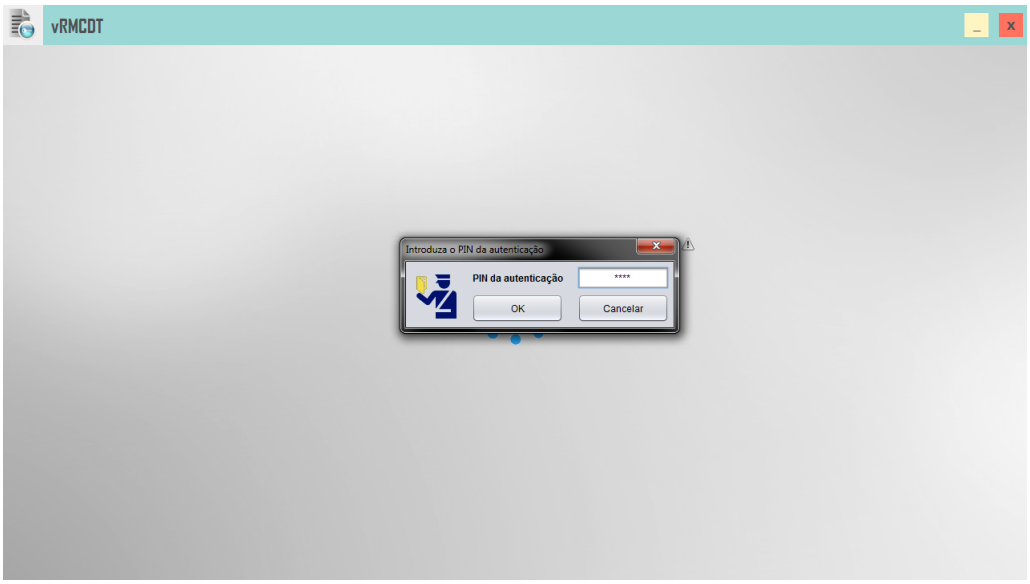


Figura 5.4: Janela de Introdução de PIN de Autenticação do Cartão de Cidadão

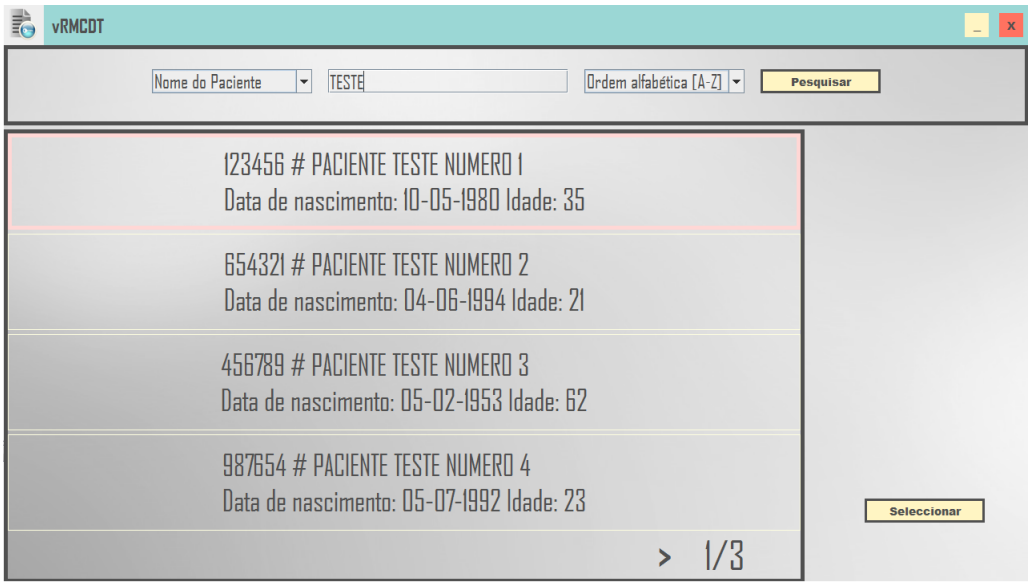


Figura 5.5: Janela de Pesquisa de Pacientes do vRMCDT

Para selecionar um utente, basta carregar na caixa correspondente e esta aparece destacada com uma cor diferente(5.5). Após esta seleção e através do botão Selecionar, é possível ver os exames do paciente tal como se visualiza na figura 5.6.

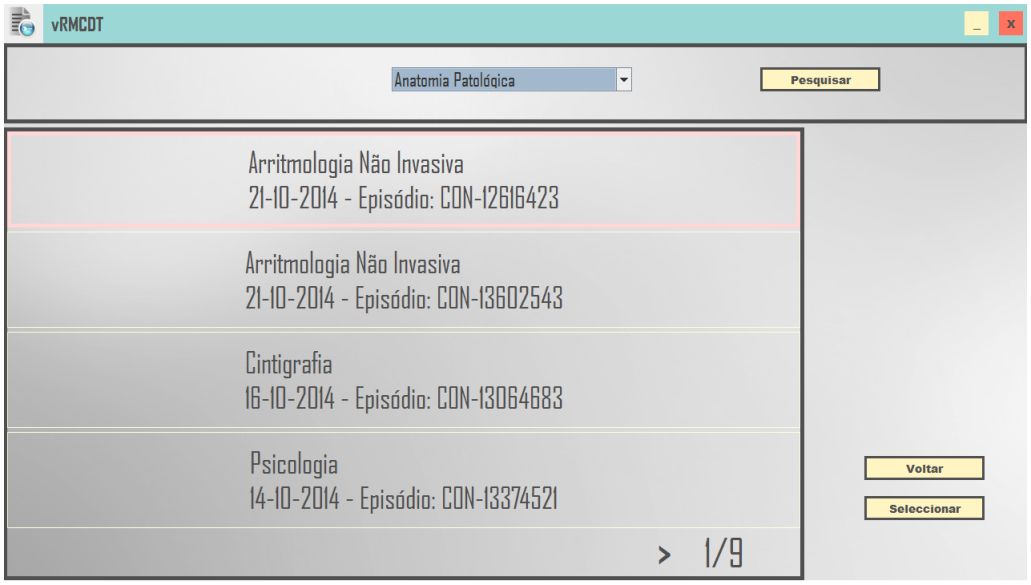


Figura 5.6: Janela de Apresentação de Exames de um Utente

A pesquisa dos exames pode ser filtrada, relativamente ao tipo de exame que se quer ver. Esta filtragem é possível através de uma pequena lista no topo da janela (5.7), sendo que nesta lista só aparecem exames que o utente tenha realizado para não serem devolvidas pesquisas nulas.

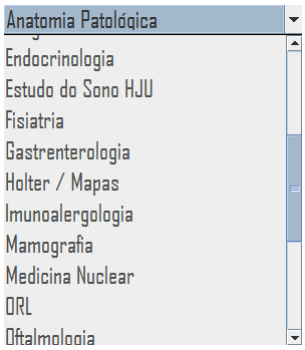


Figura 5.7: Lista para Filtragem da Pesquisa dos Exames de um Utente

Da mesma forma que selecionou um utente, o utilizador do vRMCDT deve escolher o exame que quer visualizar e carregar no botão Selecionar. Este processo leva o médico a uma janela semelhante à da figura 5.8. Nesta, são apresentados todos os detalhes relativamente a esse exame, inclusive por quem foi executado, para ser possível confirmar se é esse mesmo relatório que se quer consultar.

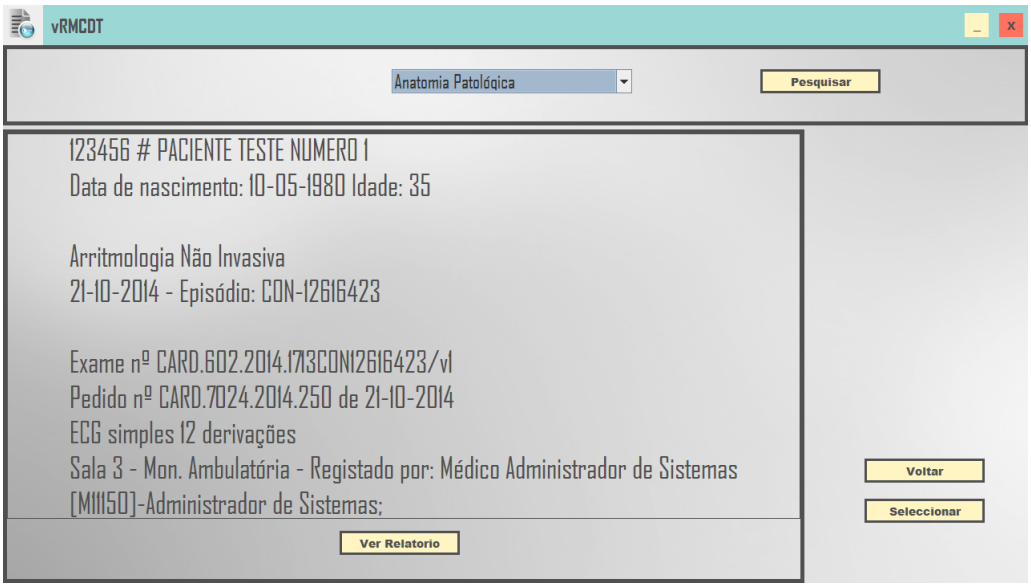


Figura 5.8: Janela de Apresentação dos Detalhes de um Exame

O botão Ver Relatório da janela da figura 5.8 faz com que apareça no ecrã uma nova janela desta vez semelhante à apresentada na figura 5.9, onde o médico pode consultar o relatório do exame em formato pdf.

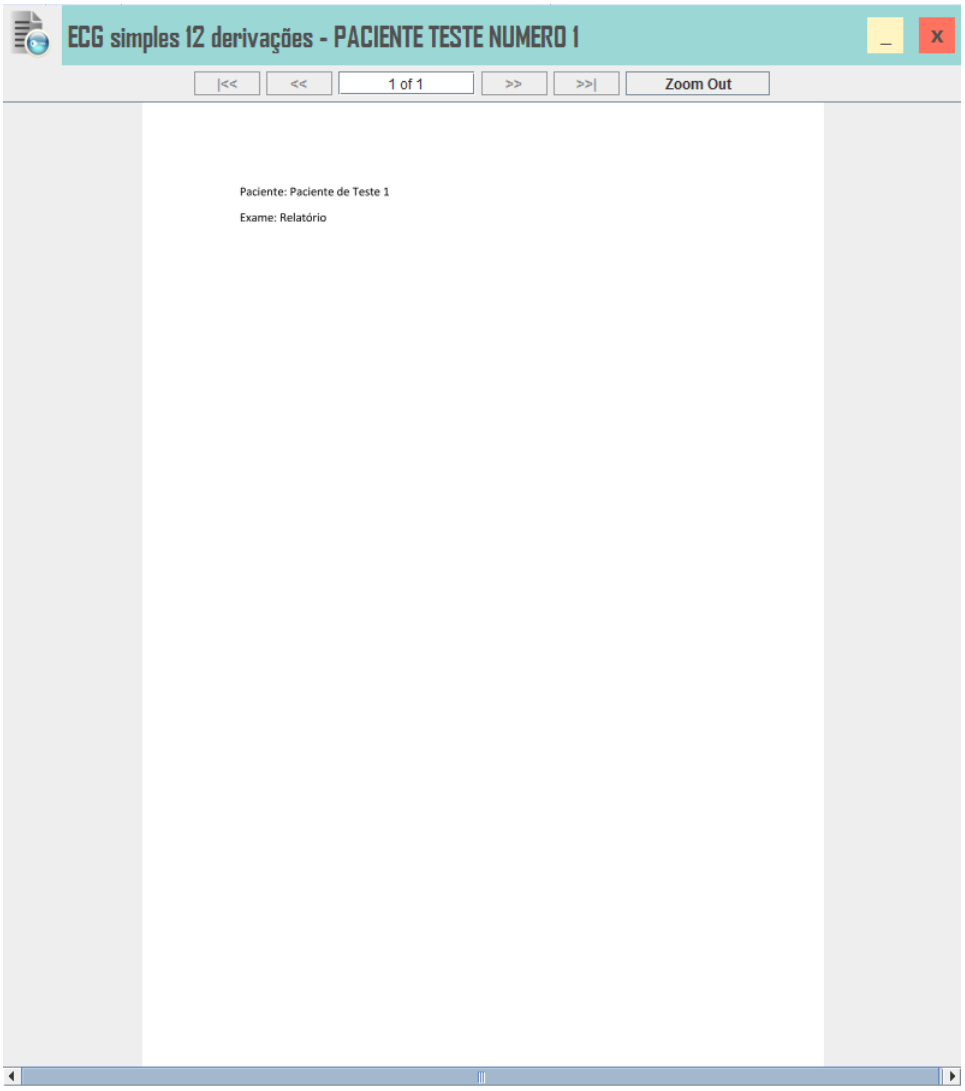


Figura 5.9: Janela de Visualização de um Relatório de um Exame Complementar de Diagnóstico e Terapêutica

5.2 Comunicação entre o vRMCDT e sRMCDT

Ao longo deste subcapítulo, são apresentados os resultados obtidos para a comunicação entre os terminais vRMCDT e sRMCDT (cliente e servidor, para facilitar a leitura). Os dados apresentados foram obtidos no terminal do cliente através da ativação do *debug* do TLS e só será mostrado o essencial,

visto ser um *output* extenso. Do lado do servidor, obter-se-ia informação complementar.

Inicialmente, o cliente adiciona como certificado confiável, o da raiz do SSARCE:

- adding as trusted cert: Subject: EMAILADDRESS=none@none.com, CN=SSARCE, OU=IT, O=CHP, L=Porto, ST=Porto, C=PT

adding as trusted cert:

De seguida, encontra uma chave privada relativa ao certificado *vrmcdt1*.

- found key for : vrmcdt1

Posteriormente, é enviado um *ClientHello* para o servidor para iniciar a comunicação com todas as características referidas no capítulo 1.1.8:

- ClientHello, TLSv1

Recebe-se, então, um ServerHello que indica quais os algoritmos a utilizar e o certificado servidor:

- ServerHello, TLSv1
- Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
- Subject: CN=srmcdt, OU=IT, O=CHP, L=Porto, ST=Porto, C=PT

Depois das verificações de certificados, são geradas todas as chaves e valores necessárias à comunicação, tal como descrito no capítulo 2.4:

- PreMaster Secret
- Client Nonce
- Server Nonce
- Master Secret
- Client MAC write Secret

- Server MAC write Secret
- Client write key
- Server write key
- Client write IV
- Server write IV

Pela análise dos algoritmos utilizados (TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA), verifica-se que tanto os algoritmos como o tamanho das chaves se coadunam com os requisitos de segurança para uma aplicação atual apresentados no capítulo 1.1.8.

5.3 Aplicação vREG

A aplicação vREG aqui apresentada já foi descrita no capítulo 4. Ao iniciar esta aplicação num *browser*, é pedido para o administrador inserir as suas credenciais de autenticação (nome de utilizador e palavra-passe), tal como na figura 5.10

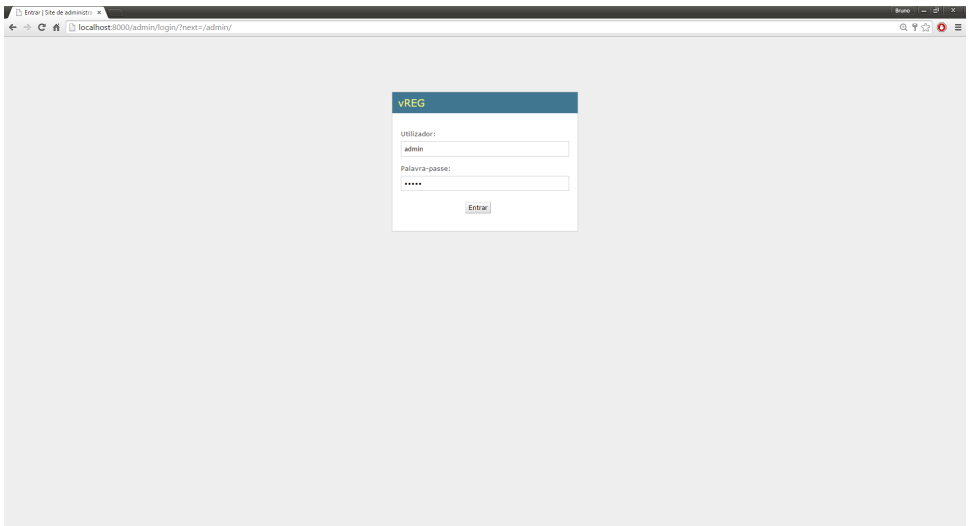


Figura 5.10: Interface de Autenticação da Aplicação vREG

Ao entrar na página, é apresentada ao administrador a interface gráfica da figura 5.11. Nesta página, pode-se aceder a outras que permitem criar novos utilizadores e gerar grupos de permissões para atribuir a cada uma dessas contas. Na mesma Interface, é possível aceder aos dados dos médicos registados no SSARCE (figura 5.12) e aos registos daquilo que cada utilizador fez ao usar o vRMCDT (figura 5.13).

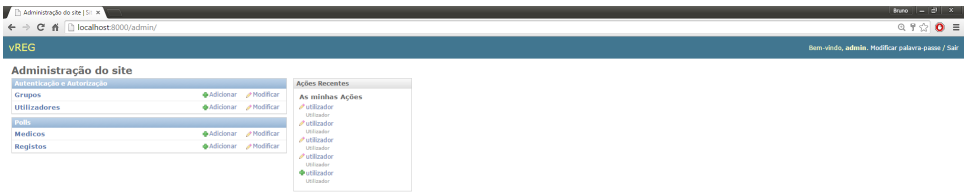


Figura 5.11: Interface Inicial da Aplicação vREG.

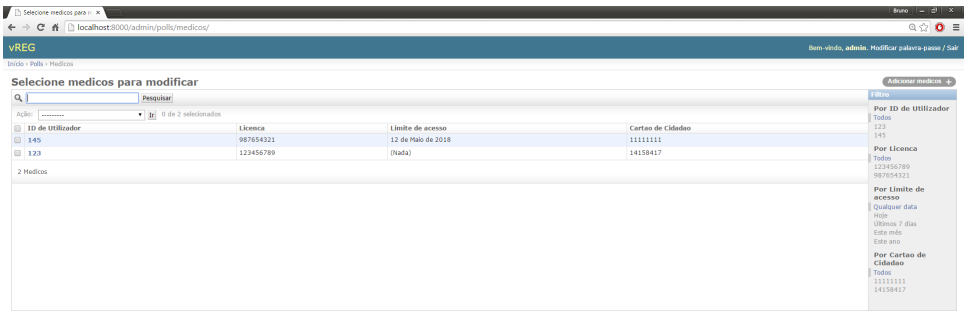


Figura 5.12: Interface da Página dos Médicos Registados no Sistema.

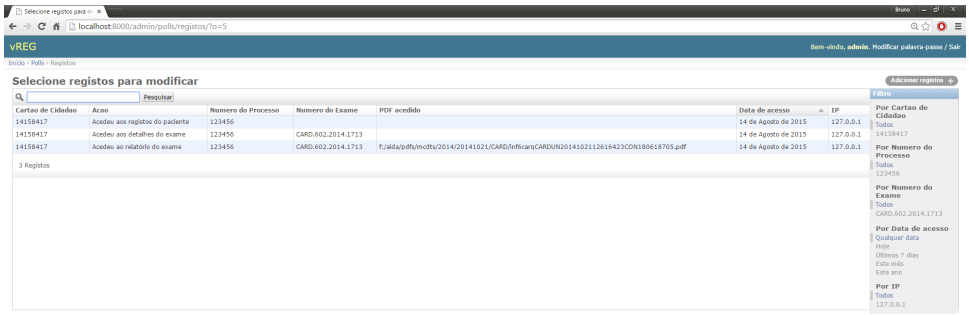


Figura 5.13: Interface dos Registos Referentes à Utilização do vRMCDT.

Na página representada na figura 5.12 e, carregando no botão Adicionar Registos +, é possível aceder a uma página para acrescentar médicos ao sistema e assim conceder permissões de acesso aos relatórios. É também possível alterar o registo de um médico já inserido na tabela, ao aceder a uma página de edição carregando no ID de utilizador do médico. Esta página de edição está representada na figura 5.14.

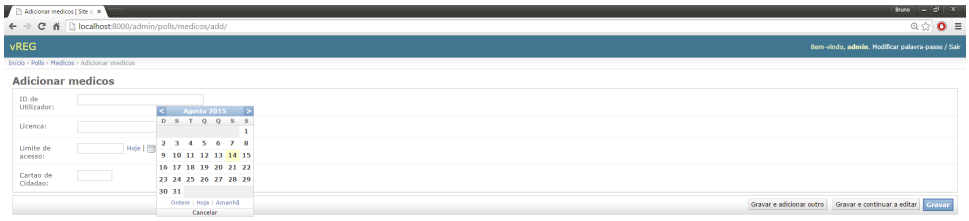


Figura 5.14: Interface de Edição do Registo de um Médico.

5.4 Análise SWOT

5.4.1 Enquadramento Teórico

Uma análise SWOT tem como objetivo identificar quer os pontos fortes (*strenghts*) quer os pontos fracos (*weaknesses*) do produto em análise. Ao mesmo tempo, procura-se descobrir quais as oportunidades (*opportunities*) e ameaças (*threats*), relativas ao ambiente em que se insere esse mesmo produto [3, 49].

A grande vantagem desta avaliação é separar o escrutínio interno da avaliação externa, referente ao ambiente do produto. Simultaneamente, é um processo bastante simples e continua a ser uma das estratégias mais utilizadas para registar os pontos fortes e fracos de uma organização [3, 49]. O objetivo da identificação destes quatro pontos é poder transformar as oportunidades em valor e anular as ameaças que possam fazer descer esse mesmo valor [3, 49].

A figura 5.15 representa a Matriz SWOT. Esta mostra a ligação entre os diferentes fatores. Os pontos fortes podem ser manipulados para encontrar oportunidades ou neutralizar ameaças. Por outro lado, os pontos fracos provocam uma situação desfavorável e dificultam o cumprimento dos objetivos do produto [3].

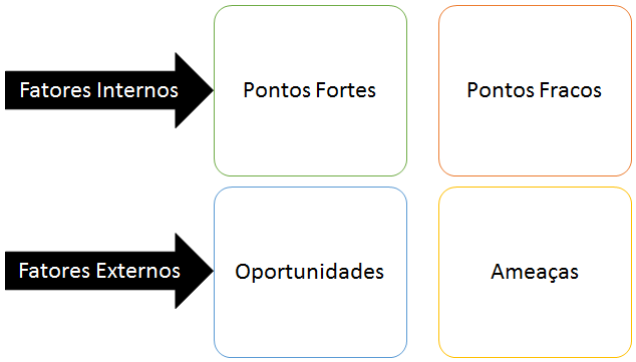


Figura 5.15: Matriz SWOT (adaptado de [3])

5.4.2 Análise SWOT do Sistema

Para implementar o que foi descrito, é necessário, então, identificar quais os pontos fortes e fracos associados ao sistema, bem como as oportunidades e ameaças a considerar.

Como pontos fortes, enaltecem-se:

- Acesso imediato ao registo clínico eletrónico para médicos registados;
- Acesso permanente ao registo clínico eletrónico;
- Facilidade de utilização com uma interface simples e intuitiva;
- Segurança personalizada para aceder aos dados, garantindo a privacidade dos doentes;
- Escalabilidade elevada;
- Possibilidade de ser usada por qualquer cidadão português;
- Acesso à informação remotamente;
- Controlo do sistema por parte de administradores numa aplicação personalizada para o efeito;
- Facilidade de adaptação a diferentes Centros de Saúde;

Contrariamente, existem, também, alguns pontos fracos a realçar:

- Requer a utilização de um leitor de cartões que teria de estar disponível em todos os terminais vRMCDT;
- Exige uma ligação à Internet;
- É necessário permitir ao sRMCDT fazer pedidos OCSP fora da Intranet do Hospital, através de uma exceção na *firewall*, levando a possíveis problemas de segurança;
- Falta de redundância de todos os componentes fundamentais ao funcionamento;

- Requer que seja garantido o acesso ao médico de forma manual por um administrador;
- Exige muito trabalho administrativo para manter os certificados atuais;
- Médicos estrangeiros não têm forma de aceder ao Sistema, visto não serem possuidores de um cartão de cidadão português.

De um outro ponto de vista, são enumeradas as oportunidades, considerados pontos fortes provenientes do ambiente em que o sistema está inserido:

- Desenvolvimento da organização;
- Aumento do número de profissionais de saúde com acesso a informação relevante;
- Recurso a novas tecnologias, aumentando a eficiência do serviço prestado;

Por fim, de uma perspetiva externa ao SSARCE, foram identificadas as ameaças ao sistema:

- Falta de aceitação por parte do Hospital por desconfiança na credibilidade dos métodos criptográficos;
- Pouca aceitação dos profissionais de saúde a mais um programa de *software*;
- Desconfiança dos médicos ao conectar o seu cartão de cidadão a um computador.

Capítulo 6

Conclusão

Após ter sido exposto todo o trabalho teórico e prático realizado durante este projeto, faz-se um balanço final de tudo o que foi desenvolvido. Por um lado, analisando se os objetivos propostos foram totalmente ou parcialmente cumpridos e quais os conselhos e sugestões para continuar e melhorar este trabalho.

Desta forma, neste capítulo, são apresentadas, na secção 6.1, as principais conclusões deste projeto e, na secção 6.2, as sugestões para melhorar o trabalho aqui relatado.

6.1 Contributos

Este projeto de dissertação tinha como objetivo inicial o desenvolvimento de um sistema que implementasse medidas de segurança e permitisse o acesso ao registo clínico eletrónico de um Hospital da área do grande Porto. Este acesso seria, para já, restrito aos relatórios de Métodos Complementares de Diagnóstico e Terapêutica e apenas autorizado a Centros de Saúde.

Existem diversos motivos para a necessidade de um sistema semelhante ao criado, já que é cada vez mais importante facilitar a consulta da informação de forma permanente e em locais que até então não possuíam tal capacidade.

Previamente à criação do sistema, foi executada uma análise de segurança. Este processo encontra-se detalhadamente descrito neste documento

e permite, por um lado, identificar quais as necessidades de segurança e quais os dados a proteger numa instituição como o CHP e, ainda, explicar de uma forma relativamente simples como é implementada essa segurança.

Esse processo é bastante importante para que sejam avaliados todos os riscos, descritas todas as suposições e se estude todas as políticas da organização. A adoção de uma nomenclatura própria facilita a leitura para indivíduos menos tecnicamente dotados. Ao mesmo tempo, permite a um indivíduo administrador verificar e compreender a forma como são protegidos os dados da sua instituição.

O sistema foi projetado de forma a permitir atualizações em diferentes módulos a fim de aumentar a segurança, sem que para isso seja necessário fazer alterações na aplicação do cliente. Este facto facilita que a aplicação esteja tecnologicamente válida constantemente.

Foi também criada uma aplicação com interface gráfica para administradores poderem gerir o sistema relativamente a médicos registados, qual o tempo que ainda dispõem para utilizar a aplicação e quais os doentes/relatórios consultados.

No término desta dissertação, todos os objetivos propostos, inicialmente, foram cumpridos, exceto alguns testes ao sistema. Porém, o sistema foi testado e revisto em termos de segurança na comunicação de dados. Infelizmente, o número de dados que se possuía para teste eram relativamente reduzidos o que não permitiu uma avaliação da *performance* do sistema. Essa verificação teria de ser feita com a aplicação conectada à Intranet do CHP que bloqueia os pedidos OCSP, não sendo possível ajuizar quanto à validade de um certificado apresentado por um médico.

6.2 Trabalho Futuro

O sistema desenvolvido durante este projeto está, atualmente, funcional para implementação nos Centros de Saúde.

Numa primeira fase e devido à não existência de testes de *performance*, aconselha-se que seja avaliada a velocidade de acesso aos dados por parte da aplicação. Esta fase colide com algumas políticas da organização, visto que

é necessário permitir o acesso aos dados por parte de uma rede exterior ao Hospital, mas é crucial para que o sistema fique funcional.

Seria, também, interessante aferir a usabilidade do sistema por parte dos profissionais de saúde nos Centros de Saúde. Esta avaliação poderia ser feita através de um inquérito que aferisse o que é considerado positivo e negativo, relativamente a facilidade de uso e compreensão das interfaces gráficas, *performance*, quantidade de acessos e aspetos a melhorar.

Devido ao facto de o vRMCDT, permitir apenas a visualização de relatórios de MCDT poderia ser também útil expandir o acesso a outras áreas do registo clínico eletrónico, começando por aquelas de mais utilidade ou mais requisitadas no inquérito proposto.

Para os administradores, uma área que disponibilizasse estatísticas relativamente a acessos e consultas poderia revelar informação importante sobre as necessidades de melhorar a aplicação e em que áreas se deveria expandir quanto à disponibilização de dados.

Por fim, seria muito pertinente a implementação de uma estratégia que permitisse o acesso ao vRMCDT por parte de médicos estrangeiros que não sejam possuidores de um cartão de cidadão.

Bibliografia

[1] W. Stallings, "Cryptography and network security: Principles and practice," 2014.

[2] N. M. A. Munassar and A. Govardhan, "A comparison between five models of software engineering," *IJCSI*, vol. 5, pp. 95–101, 2010.

[3] K. D. C. Fuscaldi, G. F. Marcelino *et al.*, "Análise swot: o caso da secretaria de política agrícola," in *SOBER. XLVI Congresso da Sociedade Brasileira de Economia, Administração e Sociologia Rural. Rio Branco, Acre*, vol. 20, 2008.

[4] T. C. Rindfleisch, "Privacy, information technology, and health care," *Communications of the ACM*, vol. 40, no. 8, pp. 92–100, 1997.

[5] F. Lopes, "Meios complementares de diagnóstico e terapêutica (mcdt) - portal da codificação clínica e dos gdh," 2000. [Online]. Available: http://portalcodgdh.min-saude.pt/index.php/Meios_Complementares_de_Diagn%C3%B3stico_e_Terap%C3%AAutica_(MCDT)#Defini.C3.A7.C3.A3o_.281.29

[6] B. Guttman and E. A. Roback, *An introduction to computer security: the NIST handbook*. DIANE Publishing, 1995.

[7] A. K. Rai, R. R. Tewari, and S. K. Upadhyay, "Different types of attacks on integrated manet-internet communication," *International Journal of Computer Science and Security*, vol. 4, no. 3, pp. 265–274, 2010.

[8] S. El Sawda and P. Urien, "Sip security attacks and solutions: A state-of-the-art review," in *Information and Communication Technologies, 2006. ICTTA'06. 2nd*, vol. 2. IEEE, pp. 3187–3191.

[9] M. Bellare and P. Rogaway, "Optimal asymmetric encryption," in *Advances in Cryptology—EUROCRYPT'94*. Springer, 1995, pp. 92–111.

[10] E. Rescorla, "Diffie-hellman key agreement method," 1999.

[11] C. Allen and T. Dierks, "The tls protocol version 1.0," 1999.

[12] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams, "Online certificate status protocol-ocsp," 1999.

[13] M. S.A., "Declaração de práticas de certificação da ec de autenticação do cartão de cidadão," Junho 2012.

[14] M. S.A., "Política de certificado de autenticação," Junho 2012.

[15] C. de Cidadão, "Manual de utilização aplicação do cartão de cidadão," Março 2015.

[16] S. K. Yoo, S. H. Kim, N. H. Kim, Y. Kang, K. Kim, S. Bae, and M. W. Vannier, "Design of a pc-based multimedia telemedicine system for brain function teleconsultation," *International journal of medical informatics*, vol. 61, no. 2, pp. 217–227, 2001.

[17] F. Jean, M. A. Kowtko, C. Yamagata, and S. Joyce, "Applying mobile application development to help dementia and alzheimer patients," 2013.

[18] V. Vaidehi, M. Vardhini, H. Yogeshwaran, G. Inbasagar, R. Bhargavi, and C. S. Hemalatha, "Agent based health monitoring of elderly people in indoor environments using wireless sensor networks," *Procedia Computer Science*, vol. 19, pp. 64–71, 2013.

[19] J. Vilaplana, F. Solsona, F. Abella, J. Cuadrado, R. Alves, and J. Mateo, "S-pc: An e-treatment application for management of smoke-quitting patients," *Computer methods and programs in biomedicine*, vol. 115, no. 1, pp. 33–45, 2014.

[20] J. E. Bardram, M. Frost, K. Szántó, M. Faurholt-Jepsen, M. Vinberg, and L. V. Kessing, "Designing mobile health technology for bipolar disorder: a field trial of the monarca system," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2013, pp. 2627–2636.

[21] R. Oliveira, S. Frutuoso, J. Machado, M. Santos, F. Portela, and A. Abelha, "Step towards m-health in pediatrics," *Procedia Technology*, vol. 9, pp. 1192–1200, 2013.

[22] S. Valente, J. Braga, J. Machado, M. Santos, and A. Abelha, "The impact of mobile platforms in obstetrics," *Procedia Technology*, vol. 9, pp. 1201–1208, 2013.

[23] S. Pereira, F. Portela, M. F. Santos, J. Machado, and A. Abelha, "Predicting preterm birth in maternity care by means of data mining," in *Progress in Artificial Intelligence: 17th Portuguese Conference on Artificial Intelligence, EPIA 2015, Coimbra, Portugal, September 8-11, 2015. Proceedings*, vol. 9273. Springer, 2015, p. 116.

[24] A. Pereira, F. Marins, B. Rodrigues, F. Portela, M. F. Santos, J. Machado, and A. Abelha, *Improving Quality of Medical Service with Mobile Health Software*. Procedia Computer Science, 2015.

[25] P. Healthcare, *PatientKeeper Architecture*, 2015. [Online]. Available: https://www.patientkeeper.com/learn_more/pk_architecture_graphic.pdf

[26] J. Herveg, F. Crazzolara, S. E. Middleton, D. Marvin, and Y. Poullet, "Gemss: Privacy and security for a medical grid," 2004.

[27] D. Solo, R. Housley, and W. Ford, "Internet x. 509 public key infrastructure certificate and crl profile," 1999.

[28] S. G. Erberich, J. C. Silverstein, A. Chervenak, R. Schuler, M. D. Nelson, and C. Kesselman, "Globus medicus-federation of dicom medical imaging devices into healthcare grids," *Studies in Health Technology and Informatics*, vol. 126, p. 269, 2007.

[29] Y. E. Gelogo and H.-K. Kim, "A design of secure e-health data management system," *Journal of Security Engineering*, vol. 12, no. 2, pp. 181–190, 2015.

[30] D. Gregorczyk, S. Fischer, T. B. Busshaus, S. Schlichting, and S. Pöhlsen, "An approach to integrate distributed systems of medical devices in high acuity environments." in *MCPS*, 2014, pp. 15–27.

[31] E. U. A. for Network and I. Security, "Algorithms, key size and parameters report - 2014," November 2014.

[32] J. Gosling, *The Java language specification*. Addison-Wesley Professional, 2000.

[33] M. Loy, R. Eckstein, D. Wood, J. Elliott, and B. Cole, *Java swing*. "O'Reilly Media, Inc.", 2002.

[34] Netbeans.org, "Netbeans ide 8.0.2 release information," 2015. [Online]. Available: https://netbeans.org/community/releases/80/

[35] Maven.apache.org, "Maven – introduction," 2015. [Online]. Available: https://maven.apache.org/what-is-maven.html

[36] GitHub, "poreid/poreid," 2015. [Online]. Available: https://github.com/poreid/poreid

[37] Technet.microsoft.com, "Chapter 1: Introduction to .net," 2015. [Online]. Available: https://technet.microsoft.com/en-us/library/bb496996.aspx

[38] Linfo.org, "Database definition," 2015. [Online]. Available: http://www.linfo.org/database.html

[39] Docs.oracle.com, "Introduction to the oracle database," 2015. [Online]. Available: http://docs.oracle.com/cd/B19306_01/server.102/b14220/intro.htm#i60746

[40] V. P. da Fonseca and F. M. Braga, "Django, desenvolvimento ágil para a web," 2009.

[41] Djangobook, "Chapter 1: Introduction to django," 2015. [Online]. Available: http://www.djangobook.com/en/2.0/chapter01.html

[42] Djangobook.com, "Chapter 2: Getting started," 2015. [Online]. Available: http://www.djangobook.com/en/2.0/chapter02.html

[43] C. C. for Information Technology, "Part 1: Introduction and general model," September 2012.

[44] C. C. for Information Technology, "Part 2: Security functional requirements," August 2005.

[45] C. C. for Information Technology, "Part 3: Security assurance components," September 2012.

[46] A. Ragen, "Manager's guide to common criteria," 2004-2013.

[47] P. Bourque and R. Fairley, "Guide to the software engineering body of knowledge version 3.0-swebok," 2014.

[48] K. Häyrinen, K. Saranto, and P. Nykänen, "Definition, structure, content, use and impacts of electronic health records: a review of the research literature," *International journal of medical informatics*, vol. 77, no. 5, pp. 291–304, 2008.

[49] A. Coman and B. Ronen, "Focused swot: diagnosing critical strengths and weaknesses," *International Journal of Production Research*, vol. 47, no. 20, pp. 5677–5689, 2009.

Apêndice A

Publicações

A.1 Systematic Coronary Risk Evaluation through Artificial Neural Networks based Systems

Autores:

Bruno Rodrigues, Sabino Gomes, Henrique Vicente, António Abelha, Paulo Novais, José Machado e José Neves

Conferência:

27th International Conference on Computer Applications in Industry and Engineering (CAINE–2014)

Ano:

2014

***Abstract*:**

On the one hand, cardiovascular diseases have severe consequences on an individual and for the society in general, once they are the main cause to death. These facts reveal that it is vital to get preventive, by knowing how

probable is to have that kind of illness. On the other hand, and until now, this risk has been assessed by a Systematic Coronary Risk Evaluation procedure that takes data from charts based on gender, age, total cholesterol, systolic blood pressure and smoking status, but with no conceivable potential to deal with the incomplete or default data that is presented on those tools. Therefore, the focus in this work will be on the development of a risk evaluation support system based on a low-risk record, grounded on a new approach to knowledge representation and reasoning, that based on an extension to the Logic Programming language, will be able to overcome the drawbacks of the present ones. This will be complemented with a computational framework based on Artificial Neural Networks.

***Keywords*:**

Systematic Coronary Risk Evaluation; Knowledge Representation and Reasoning; Logic Programming; Artificial Neural Networks.

Estado:

Publicado

A.2 Improving Quality of Medical Service with Mobile Health Software

Autores:

Ana Pereira, Fernando Marins, Bruno Rodrigues, Filipe Portela, Manuel Filipe Santos, José Machado, Fernando Rua, Álvaro Silva, António Abelha.

Conferência:

The 5th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH 2015)

Ano:

2015

***Abstract*:**

An increasing number of m-Health applications are being developed benefiting health service delivery. In this paper, a new methodology based on the principle of calm computing applied to diagnostic and therapeutic procedure reporting is proposed. A mobile application was designed for the physicians of one of the Portuguese major hospitals, which takes advantage of a multi-agent interoperability platform, the Agency for the Integration, Diffusion and Archive (AIDA). This application allows the visualization of inpatients and outpatients medical reports in a quicker and safer manner, in addition to offer a remote access to information. This project shows the advantages in the use of mobile software in a medical environment but the first step is always to build or use an interoperability platform, flexible, adaptable and pervasive. The platform offers a comprehensive set of services that restricts the development of mobile software almost exclusively to the mobile user interface design. The technology was tested and assessed in a real context by intensivists.

***Keywords*:**

m-Health application; Healthcare; Healthcare Professionals; Healthcare Quality; Clinical Information, INTCare, Calm Computing, Pervasive Systems, Ubiquitouse.

Estado:

Aceite para publicação

Apêndice B

Glossário

Namespace Conjunto de símbolos usados para organizar objetos, para que sejam referidos por esse nome (em computação).. 31

browser Aplicação de *software* que serve para pedir, enviar e apresentar informações da *World Wide Web*. Comummente descrito como aplicação que permite aos utilizadores acederem a páginas na *Internet*.. 82

firewall Sistema de proteção de uma rede ou sistema, para impedir acessos não autorizados à mesma.. 35, 86

middleware *Software* que liga componentes de *software* a aplicações de uma entidade diferente.. 22

standard Algo considerado padrão ou um modelo aprovado.. 10

aplicação distribuída Aplicações que funcionam em múltiplos computadores numa rede ao mesmo tempo.. 2, 3

Intranet Rede privada geralmente numa organização para acesso apenas do *staff* da mesma.. 18, 35, 48, 54, 57, 86, 90